



**TUGAS AKHIR – TF 145565**

**SISTEM PENGENDALIAN TEGANGAN DENGAN  
PENGATURAN LAJU ALIRAN NOSEL PADA  
PLTMH SKALA LABORATORIUM**

**Nila Dia Citra Effendy  
NRP 105 115 00000 017**

**Dosen Pembimbing I  
Hendra Cordova, S.T., M.T.  
NIP. 19690530 199412 1 001**

**Dosen Pembimbing II  
Sefi Novendra Patrialova, S.Si., M.T.  
NPP. 1991201712053**

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI  
DEPARTEMEN TEKNIK INSTRUMENTASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018**





## **TUGAS AKHIR – TF 145565**

# **SISTEM PENGENDALIAN TEGANGAN DENGAN PENGATURAN LAJU ALIRAN NOSEL PADA PLTMH SKALA LABORATORIUM**

Nila Dia Citra Effendy  
NRP 105 115 00000 017

Dosen Pembimbing I  
Hendra Cordova, S.T., M.T.  
NIP. 19690530 199412 1 001

Dosen Pembimbing II  
Sefi Novendra Patrialova, S.Si., M.T.  
NPP. 1991201712053

PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI  
DEPARTEMEN TEKNIK INSTRUMENTASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



***FINAL PROJECT – TF 145565***

***VOLTAGE CONTROL SYSTEM WITH NOZZLE FLOWRATE  
CONTROL ON PLTMH LABORATORY SCALE***

Nila Dia Citra Effendy  
NRP 105 115 00000 017

*Supervisor I*  
Hendra Cordova, S.T., M.T.  
NIP. 19690530 199412 1 001

*Supervisor II*  
Sefi Novendra Patrialova, S.Si., M.T.  
NPP. 1991201712053

***DIII INSTRUMENTATION TECHNOLOGY  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
Faculty of Vocations  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018***

**SISTEM PENGENDALIAN TEGANGAN DENGAN  
PENGATURAN LAJU ALIRAN NOSEL PADA PLTMH  
SKALA LABORATORIUM**

**TUGAS AKHIR**

**Oleh :**

**Nila Dia Citra Effendy  
NRP. 105 115 00000 017**

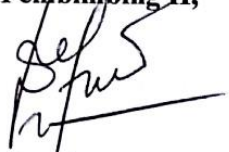
**Surabaya, 25 Juli 2018  
Mengetahui dan Menyetujui,**

**Dosen Pembimbing I,**



**Hendra Cordova, S.T., M.T.  
NIP. 19690530 199412 1 001**

**Dosen Pembimbing II,**



**Sefi Novendra Patrialova, S.Si., M.T.  
NPP. 1991201712053**

**Kepala Departemen Teknik  
Instrumentasi,**



**Dr. Purwadi Agus Darwito, M.Sc.  
NIP. 19620822 198803 1 001**

**SISTEM PENGENDALIAN TEGANGAN DENGAN  
PENGATURAN LAJU ALIRAN NOSEL PADA PLTMH  
SKALA LABORATORIUM**


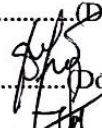

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar  
Ahli Madya  
Pada Program Studi DIII Teknologi Instrumentasi  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

**Oleh :**

**Nila Dia Citra Effendy  
NRP. 105 115 00000 017**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Hendra Cordova, S.T., M.T..... (Dosen Pembimbing I)
2. Sefi Novendra Patrialova, S.Si., M.T..... (Dosen Pembimbing II)
3. Ahmad Fauzan Adziimaa, S.T., M.Sc..... (Dosen Penguji )

**SURABAYA  
JULI 2018**

# **SISTEM PENGENDALIAN TEGANGAN DENGAN PENGATURAN LAJU ALIRAN NOSEL PADA PLTMH SKALA LABORATORIUM**

**Nama Mahasiswa** : Nila Dia Citra Effendy  
**NRP** : 105 115 00000 017  
**Departemen** : Teknik Instrumentasi FV-ITS  
**Dosen Pembimbing I** : Hendra Cordova, S.T., M.T.  
**Dosen Pembimbing II** : Sefi Novendra Patrialova, S.Si., M.T.

## **ABSTRAK**

Pembangkit Listrik Tenaga Mikro Hidro (PLTMH) merupakan salah satu energi alternatif yang terdapat di Indonesia. Namun, selama ini banyak PLTA kelas mikrohidro tidak berfungsi dengan maksimal. Salah satu masalahnya adalah dari pasokan air atau pengendalian *flow* pada sistem PLTMH itu sendiri. Sehingga, berpengaruh pada jumlah putaran generator yang menjadi *overload* dan besar tegangan yang dihasilkan tidak stabil. Pada perancangan sistem pengendalian tegangan dengan pengaturan laju aliran ini, sensor yang digunakan sensor tegangan untuk mendeteksi tegangan yang dihasilkan oleh generator dan *water flow sensor* digunakan untuk mendeteksi debit aliran yang melalui pipa. *Output* dari sensor tegangan akan diterima oleh ATmega128 dan *motor valve* akan memberikan respon sesuai perintah *controller*. Dari data respon pengendalian yang didapat menunjukkan bahwa pengendalian telah berjalan sesuai dengan *setpoint* yang diinginkan yaitu 38 volt untuk menyalakan lampu dan beresilasi mendekati setpoint. Selain itu berdasarkan pengujian alat pada sistem pengendalian tegangan dengan pengaturan laju aliran didapatkan nilai *rise time* sebesar 24 detik, nilai *maximum overshoot* sebesar 11,24%, dan *settling time* selama 22 detik.

**Kata Kunci** : PLMH, Tegangan, Pengendalian

## ***VOLTAGE CONTROL SYSTEM WITH NOZZLE FLOWRATE CONTROL ON PLTMH LABORATORY SCALE***

**Name of Student** : Nila Dia Citra Effendy  
**NRP** : 105 115 00000 017  
**Departement** : Instrumentation Engineering FV-ITS  
**Supervisor I** : Hendra Cordova, S.T., M.T.  
**Supervisor II** : Sefi Novendra Patrialova, S.Si., M.T.

### ***ABSTRACT***

*Micro Hydro Power Plant is one of the alternative energy in Indonesia. However, there are many micro-hydro class hydropower do not function optimally. One of the problem is from water supply or flow control in the system. So, it affects the rotation per minutes of the generator that becomes overloaded and the amount of voltage generated is unstable. The voltage control system with setting by flow rate, the sensor used are voltage sensor to detect the voltage from generator and water flow sensor to detect the flow rate. The output of the voltage sensor will be received by the ATmega128 and the motor valve will respond according to the controller's command. From the control response data obtained shows that the control has run according to the desired setpoint which is 38 volts to turn on the lamp and oscillate close to the setpoint. Besides that, based on the tool testing on the voltage control system with flow rate regulation, the rise time value is 24 seconds, the maximum overshoot value is 11.24%, and the settling time is 22 seconds.*

***Keywords*** : PLTMH, voltage, control



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“SISTEM PENGENDALIAN TEGANGAN DENGAN PENGATURAN LAJU ALIRAN NOSEL PADA PLTMH SKALA LABORATORIUM”** dengan tepat waktu. terselesaikannya laporan ini juga tak luput dari dukungan berbagai pihak. Untuk itulah dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr .Ir. Purwadi Agus D, M.Sc. selaku Kepala Departemen Teknik Instrumentasi FV-ITS.
2. Bapak Hendra Cordova, S.T., M.T. dan Ibu Sefi Novendra Patrialova, S.Si., M.T. selaku pembimbing Tugas Akhir yang telah membina dengan baik dan sabar.
3. Bapak Jerry Susatio, M.T. selaku Dosen Wali penulis.
4. Kedua orang tua yang tidak henti-hentinya memberi semangat dan doa untuk menyelesaikan Tugas akhir ini
5. PLMH *Team* (Ragil, Delima, Arifur, Hanif, dan Agam) yang telah membantu pengerjaan Tugas Akhir ini hingga selesai.
6. Teman-teman F50 khususnya DIII Teknologi Instrumentasi angkatan 2015 FV-ITS.
7. Serta semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih kurang sempurna. Oleh karena itu penulis menerima segala masukan baik berupa saran, kritik demi kesempurnaan lapiran ini. Demikian laporan Tugas Akhir ini ditulis dengan harapan dapat bermanfaat.

Surabaya, 25 Juli 2018

Penulis





## DAFTAR ISI

	Hal
<b>HALAMAN JUDUL.....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN.....</b>	<b>iii</b>
<b>ABSTRAK.....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>

### **BAB I. PENDAHULUAN**

1.1 Latar Belakang .....	1
1.2 Permasalahan.....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah.....	2
1.5 Manfaat.....	3

### **BAB II. DASAR TEORI**

2.1 Mikrohidro .....	5
2.2 Sistem, Prinsip Kerja dan Komponen Mikrohidro .....	5
2.3 Studi Literatur .....	7
2.4 Konsep Aliran Air .....	8
2.4.1 Persamaan Kontinuitas .....	8
2.4.2 Persamaan Bernouli .....	8
2.5 Sistem Kontrol.....	9
2.6 Prinsip Kerja Motor Valve .....	9
2.7 Prinsip Kerja Sensor Tegangan DC.....	13

### **BAB III. PERANCANGAN DAN PEMBUATAN ALAT**

3.1 Bahan dan Peralatan yang Digunakan .....	15
3.2 Metodologi Pengerjaan.....	18
3.2.1 <i>Flowchart</i> Perancangan Alat .....	18

3.3 Perancangan Alat Tugas Akhir .....	21
3.3.1 Perancangan PFD, dan P&ID .....	24
3.4 Perancangan Perangkat Lunak ( <i>Software</i> ) .....	27
3.4.1 <i>Code Vision AVR V2.05.0</i> .....	27
3.4.2 Khazama V 1.7.0 .....	27

## **BAB IV. HASIL DAN PEMBAHASAN**

4.1 Rancang Bangun Alat .....	29
4.1.1 Perancangana Sistem Pengendalian Tegangan .....	29
4.1.2 Pengujian Komponen .....	30
4.1.2.1 Pengujian Pompa Air .....	30
4.1.2.2 Pengujian Motor Valve .....	32
4.1.2.2 Pengujian Sensor Tegangan .....	34
4.2 Pengujian Sistem .....	38
4.3 Pembahasan .....	42

## **BAB V. PENUTUP**

5.1 Kesimpulan.....	45
5.2 Saran.....	45

## **DAFTAR PUSTAKA**

**LAMPIRAN A** (*Listing Program Pada Code Vision Avr*)

**LAMPIRAN B** (*Data Sheet System Atmega128*)

**LAMPIRAN C** (*Data Sheet Water Flow Sensor Type FS400A*)

## **BIODATA PENULIS**

## DAFTAR GAMBAR

	Hal
<b>Gambar 2.1</b> Komponen Umum PLTMH .....	5
<b>Gambar 2.2</b> Motor stepper dengan lilitan unipolar.....	10
<b>Gambar 2.3</b> Motor stepper dengan lilitan Bipolar.....	11
<b>Gambar 2.4</b> Rangkaian Pembagi Tegangan .....	13
<b>Gambar 3.1</b> Pompa Air .....	15
<b>Gambar 3.2</b> <i>Motor Valve</i> .....	15
<b>Gambar 3.3</b> <i>Voltage Sensor</i> .....	16
<b>Gambar 3.4</b> <i>Water Flow Sensor</i> FS400A G 1”.....	16
<b>Gambar 3.5</b> Mikrokontroler ATmega 128.....	17
<b>Gambar 3.6</b> <i>Driver Motor L298N</i> .....	17
<b>Gambar 3.7</b> <i>Flowchart</i> Pengerjaan Tugas Akhir .....	19
<b>Gambar 3.8</b> Diagram Blok Pengendalian .....	20
<b>Gambar 3.9</b> Diagram Kerja Sistem.....	21
<b>Gambar 3.10</b> Diagram Alir Integrasi .....	22
<b>Gambar 3.11</b> Diagram Alir pengendalian Motor Valve .....	23
<b>Gambar 3.12</b> <i>Process Flow Diagram</i> .....	24
<b>Gambar 3.13</b> <i>Piping and Instrument Diagram</i> .....	25
<b>Gambar 3.14</b> <i>Design Plant PLTMH</i> .....	26
<b>Gambar 3.15</b> Tampilan <i>Software CV AVR V 2.05.0</i> .....	27
<b>Gambar 3.16</b> Khazama V 1.7.0.....	28
<b>Gambar 4.1</b> Rancang Bangun PLTMH.....	29
<b>Gambar 4.2</b> Sistem Pengendalian tegangan dan laju aliran....	29
<b>Gambar 4.3</b> Lampu LED .....	30
<b>Gambar 4.4</b> Rangkaian motor valve .....	32
<b>Gambar 4.5</b> Grafik Pembacaan oleh Sensor dan Multimeter..	36
<b>Gambar 4.6</b> Grafik Uji Sistem .....	41

## DAFTAR TABEL

	Hal
<b>Tabel 2.1</b> Pergerakan Half Step .....	12
<b>Tabel 2.2</b> Pergerakan Full Step .....	12
<b>Tabel 4.1</b> Hasil Pengujian Pompa.....	31
<b>Tabel 4.2</b> Hasil Pengujian <i>Motor Valve</i> .....	33
<b>Tabel 4.3</b> Hasil Pengambilan Data Naik.....	34
<b>Tabel 4.4</b> Hasil Pengambilan Data Turun.....	35
<b>Tabel 4.5</b> Data Kalibrasi Sensor .....	35
<b>Tabel 4.6</b> Data Kalibrasi Sensor Lanjutan .....	36
<b>Tabel 4.7</b> Hasil Pengujian Sistem .....	39







# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Penggunaan listrik merupakan faktor yang penting dalam kehidupan, baik pada sector rumah tangga, penerangan, komunikasi industri dan sebagainya. Hampir seluruh aktifitas yang dilakukan manusia membutuhkan listrik. Dewasa ini pembangkit listrik tenaga uap (PLTU) sudah bisa tergantikan oleh berbagai energi terbarukan, salah satunya adalah PLTA atau pembangkit listrik tenaga air, yang prinsipnya menggunakan air sebagai tenaga penggerakannya dengan cara memanfaatkan tinggi terjunan dan jumlah debit air per detik yang dipicu dengan pompa sehingga akan memutar poros turbin dan menghasilkan energi mekanik. Energi ini selanjutnya menggerakkan generator dan menghasilkan listrik (Wirabudi, 2017). Selama ini banyak PLTA kelas mikrohidro tidak berfungsi dengan maksimal, dan salah satu masalahnya adalah dari *suplay* air atau pengendalian *flow* pada sistem PLTMH itu sendiri sehingga berpengaruh pada jumlah putaran generator dan besar tegangan yang dihasilkan tidak stabil (Maulana and Dkk, 2017).

Pada penelitian ini memanfaatkan energi potensial dan energi kinetik air sebagai penggerak turbin, dimana konsep aliran air yang dipakai digantikan dengan menggunakan pompa air (Winanti, 2014). Pada sistem ini, sangat rentan dengan kondisi aliran air yang berubah-ubah. Untuk menjaga putaran turbin dan tegangan yang dihasilkan generator agar tetap stabil dan tidak terjadi *overload* pada generator, maka dibutuhkan sistem pengendalian aliran yang aman agar dapat mengurangi resiko dan mempermudah pengaturan debit aliran air. Pengaturan debit air menggunakan motor valve yang dapat diatur dengan posisi tertentu secara otomatis sesuai dengan kondisi air (Abdur Rohman and Dkk, 2017).

Dalam rangka pengerjaan tugas akhir yang kemudian disusul dengan adanya peluang bagi mahasiswa untuk turut serta dalam mengembangkan departemen dengan memberikan

kontribusi dalam skala laboratorium, maka dirancang Sistem Pengendalian Tegangan dengan Pengaturan laju Aliran Nosel Pada Pembangkit Listrik Tenaga Mikrohidro (PLTMH) skala laboratorium dengan meningkatkan efektifitas debit aliran air untuk menghasilkan jumlah putaran turbin yang stabil . Penelitian ini juga dapat digunakan sebagai praktikum beberapa mata kuliah seperti sistem pengukuran ( listrik, aliran, ), elektronika industri, program komputer, fisika terapan, teknik otomasi, dan sistem instrumentasi industri.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah :

1. Bagaimana cara membangun sebuah sistem pengendalian tegangan dengan pengaturan aliran nosel pada PLTM,
2. Bagaimana mengontrol tegangan dengan mengatur sistem aliran nosel pada PLTMH.

### **1.3. Tujuan**

Tujuan dari rancang bangun alat ini adalah untuk memberikan solusi pada rumusan masalah yaitu :

1. Untuk membangun sebuah sistem pengendalian tegangan dengan pengaturan aliran nosel pada PLTM,
2. Untuk mengontrol tegangan dengan mengatur sistem aliran nosel pada PLTMH.

### **1.4. Batasan Masalah**

Adapun batas ruang lingkup dari penelitian tugas akhir ini sebagai berikut :

1. Mikrokontroler yang digunakan adalah 128,
2. Variabel yang dikontrol adalah tegangan yang dihasilkan oleh generator.

3. Variabel yang dimanipulasi adalah laju aliran yang melalui motor valve.

### **1.5. Manfaat**

Manfaat dari tugas akhir ini adalah sebagai berikut :

1. Sebagai sistem pengendalian aliran nosel pada PLTMH agar mendapatkan debit aliran air yang sesuai dan stabil sehingga dapat memutar turbin dan menghasilkan tegangan yang stabil.
2. Sebagai penunjang praktikum bagi mahasiswa Teknik Instrumentasi pada beberapa mata kuliah seperti sistem pengukuran ( listrik, aliran, ), program komputer, fisika terapan, teknik otomasi, dan sistem instrumentasi industri.

*(Halaman Sengaja Dikosongkan)*

## **BAB II**

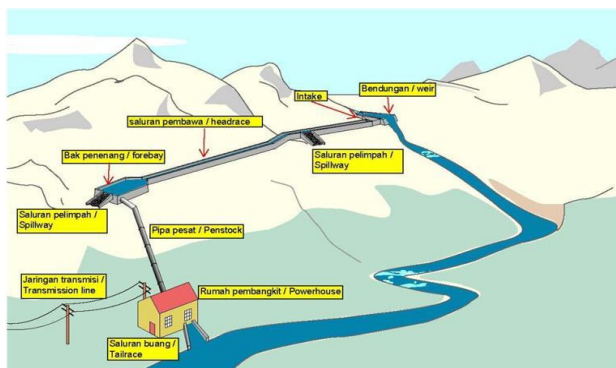
### **DASAR TEORI**

#### **2.1 Mikrohidro**

Pembangkit Listrik Tenaga Mikrohidro (PLMTH), adalah pembangkit listrik dalam skala mikro yang menggunakan air sebagai tenaga penggeraknya untuk memutar turbin, dengan cara memanfaatkan beda ketinggiannya dan jumlah debit air yang dipicu dengan pompa dan akan memutar poros turbin sehingga menghasilkan energi kinetik. Energi ini selanjutnya akan menggerakkan generator dan menghasilkan listrik. Secara teknis, mikrohidro memiliki tiga komponen utama yaitu air, turbin dan generator. Mikrohidro mendapatkan energi dari aliran air yang memiliki perbedaan ketinggian tertentu (Wirabudi, 2017) .

Secara khusus mikrohidro mengacu pada sistem yang menghasilkan tenaga pada skala 5kW sampai 100kW. Pada dasarnya, mikrohidro memanfaatkan energi potensial jatuhnya air (*head*). Semakin tinggi jatuhnya air maka semakin besar energi potensial air yang dapat diubah menjadi energi listrik (Pasalli and Rehiara, 2014).

#### **2.2 Sistem, Prinsip Kerja dan Komponen Mikrohidro**



**Gambar 2. 1** Komponen umum PLTMH

(Sumber : <https://www.kajianpustaka.com/2016/10/pembangkit-listrik-tenaga-mikro-hidro.html>)

Gambar diatas menunjukkan komponen umum dan sistem sistem mikrohidro. Komponen PLTMH ini terdiri dari bendungan (*storage*), *water intake*, saluran pelimpah, saluran pembawa, bak penenang (*forebay*), saluran pelimpah, pipa pesat (*penstock*), rumah pembangkit (*powerhouse*), dan saluran pembuangan. Dasar dari perencanaan PLTMH dimulai dari penentuan lokasi intake, aliran air yang akan sampai ke turbin (Pasalli and Rehiara, 2014).

Aliran air masuk dari hulu akan dialirkan pada *water intake* dan selanjutnya dialirkan melalui saluran pembawa. Saluran pembawa tersebut mengalirkan air menuju bak penenang (*forebay*) sebelum menuju ke pipa *penstock*. Di dalam bak penenang, air akan di saring pada saluran pelimpah. Setelah disaring, air akan dialirkan melaui pipa pesat menuju rumah pembangkitan. Konversi daya dilakukan di dalam rumah pembangkit, dan turbin akan memindahkan Energi mekanik ke generator, kemudian generator menghasilkan energi listrik (Trisasiwi, 2017).

Sebuah skema hidro memerlukan dua hal yaitu debit air dan ketinggian jatuh (biasa disebut '*Head*') untuk menghasilkan energi. Menkonversi energi potensial dari bentuk ketinggian dan energi kinetik aliran air , dan diubah menjadi energi listrik (Ceri Steward Poea, 2013). Persamaan untuk perhitungan daya adalah :

$$P = p \times g \times Q \times H_{eff} \dots\dots\dots(2.1)$$

Dimana :

P = Daya (Watt)

p = massa jenis air ( $kg/m^3$ )

g = gravitasi bumi ( $m/s^2$ )

Q = Debit aliran air ( $m^3/s$ )

$$H_{eff} = H_{gross} - Losses \dots\dots\dots(2.2)$$

Dimana :

H<sub>gross</sub> = Head kotor (m)

Losses = Head loss (m)

Selain memanfaatkan jatuhnya air, PLTMH juga dapat diperoleh dari aliran air datar. Energi yang dapat digunakan langsung, yaitu energi kinetik air (Khalid, 2016).

$$P = \frac{1}{2} \rho Q v^2 \quad (2.3)$$

Dimana :

P = Daya (Watt)

$\rho$  = massa jenis air ( $kg/m^3$ )

v = kecepatan aliran air ( $m/s$ )

Q = Debit aliran air ( $m^3/s$ )

### 2.3 Studi Literatur

Mempelajari literatur dari penelitian – penelitian yang pernah dilakukan sebelumnya. Pada tahun 2012, oleh Fifi Hesty Sholihah yaitu Rancang Bangun Prototipe Pembangkit Listrik Tenaga Mikrohidro. Penelitian ini hanya memanfaatkan beda ketinggian 1,3 m dan debit 2 lt/s. Namun, daya dan tegangan yang dihasilkan sangat kecil secara berurutan yaitu sebesar 0,24 Watt dan 2,36 volt. Tahun 2013, oleh (Firmansyah, 2013) yaitu Distribusi Tekanan Fluida pada Nozel Turbin Pelton Berskala Mikro dengan Menggunakan Perangkat Lunak *Solidworks*. Penelitian ini menggunakan *manual valve* sehingga sistem yang digunakan kurang efisien. Pada tahun 2014, oleh Naftalin Winanti yaitu Prototipe PLTA dengan Memanfaatkan Energi Kinetik Air untuk Penerangan. Pada penelitian menggunakan konsep aliran datar yang dipompa menuju turbin. Namun, pada penelitian ini menghasilkan tegangan yang sangat kecil (Winanti, 2014). Pada tahun 2016 penelitian ini juga dilakukan oleh Wahyu Idang Wirabudi yaitu Pengembangan Sistem Kontrol Gate pada *Flow Control Valve* Sistem Pembangkit Daya Mikrohidro Berbasis PLC Siemens S7-1200 (Wirabudi, 2017). Pada penelitian ini sistem kendali yang digunakan adalah menggunakan PLC sehingga harganya akan lebih tinggi dibandingkan jika dengan menggunakan mikrokontroler. Pada tahun 2017, penelitian oleh Fahmi Arif Maulana yaitu Prototipe Kontrol Aliran Air Berbasis Mikrokontroler untuk Pembangkit Listrik Tenaga Mikrohidro.



Pada penelitian ini pengontrolan air hanya pada tangki air atau bak penampungnya saja tanpa adanya pengontrolan debit aliran air yang melalui katup pengendali (*valve*) menuju ke turbin (Maulana and Dkk, 2017).

## 2.4 Konsep Aliran Air

Air yang mengalir mempunyai energi. Air yang berada pada beda ketinggian memiliki energi potensial yang dapat dimanfaatkan untuk menggerakkan sudu-sudu turbin, sehingga turbin dapat memutar generator yang akan menghasilkan listrik. Besarnya aliran sangat bergantung pada jumlah debit air. Selain dipengaruhi debit, energi air juga dapat dipengaruhi oleh keadaan lingkungan air seperti ukuran pipa yang dialiri air dan bentuk aliran air. Semakin besar debit yang dialirkan, maka semakin besar daya yang diperoleh (Sukamta and Kusmantoro, 2015).

### 2.4.1 Persamaan Kontinuitas

Persamaan kontinuitas merupakan hubungan antara laju atau kecepatan aliran dan luas penampang yang dilalui aliran fluida seperti pada persamaan dibawah ini :

$$v_1 \times A_1 = v_2 \times A_2 \dots \dots \dots (2.4)$$

Dimana :

$v_1$  = kecepatan pada kondisi awal ( $m/s$ )

$A_1$  = Luas penampang 1 ( $m^2$ )

$v_2$  = kecepatan pada kondisi kedua ( $m/s$ )

$A_2$  = Luas penampang 2 ( $m^2$ )

Pada persamaan diatas menunjukka bahwa laju aliran meningkat ketika kita mengurangi luas penampang dimana fluida mengalir (Sofyana, 2017).

### 2.4.2 Persamaan Bernouli

Persamaan bernouli adalah penerapan dari prinsip kekekalan energi mekanik pada aliran fluida, yang menghasilkan persamaan sebagai berikut :

$$P_1 + \frac{1}{2}pv_1^2 + pgh_1 = P_2 + \frac{1}{2}pv_2^2 + pgh_2 \dots \dots \dots (2.5)$$

Dimana :

P = Tekanan (Pa)

p = Massa jenis ( $kg/m^3$ )

v = Kecepatan ( $m/s$ )

h = Ketinggian (m)

## 2.5 Sistem Kontrol

Pengertian sistem kontrol itu sendiri adalah proses pengaturan/pengendalian terhadap satu atau beberapa besaran (variabel, parameter) sehingga berada pada suatu harga atau dalam suatu rangkuman harga (*range*) tertentu. Dalam istilah lain disebut juga teknik pengaturan, sistem pengendalian atau sistem pengontrolan.

Sistem kontrol (*control system*) merupakan suatu kumpulan cara atau metode yang dipelajari dari kebiasaan-kebiasaan manusia dalam bekerja, dimana manusia membutuhkan suatu pengamatan kualitas dari apa yang telah mereka kerjakan sehingga memiliki karakteristik sesuai dengan yang diharapkan pada mulanya. Perkembangan teknologi menyebabkan manusia selalu terus belajar untuk mengembangkan dan mengoperasikan pekerjaan-pekerjaan kontrol yang semula dilakukan oleh manusia menjadi serba otomatis (dikendalikan oleh mesin) (Dr. Aris Triwiyatno, 2008).

Dalam aplikasinya, sistem kontrol memegang peranan penting dalam teknologi. Sebagai contoh, otomatisasi industri dapat menekan biaya produksi, mempertinggi kualitas, dan dapat menggantikan pekerjaan-pekerjaan rutin yang membosankan. Sehingga dengan demikian akan meningkatkan kinerja suatu sistem secara keseluruhan, dan pada akhirnya memberikan keuntungan bagi manusia yang menerapkannya.

## 2.6 Prinsip Kerja Motor Valve

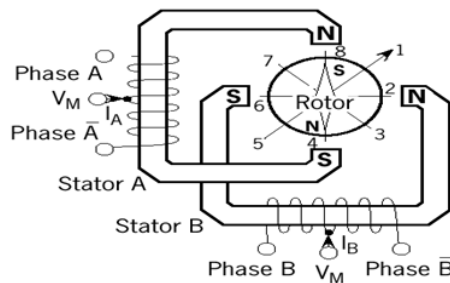
Motor Valve merupakan katup yang digerakkan oleh motor yang dikombinasikan dengan *gear-gear* dan dipicu oleh pulsa

elektrik yang ada pada motor tersebut. Kemudian untuk menstop motor yang bergerak dipasang limit switch (sebagai pemutus power ke motor).

Salah satu jenis *motor control valve* dc yang dikendalikan dengan pulsa-pulsa digital. Prinsip kerja motor stepper adalah bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit dimana motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor stepper tersebut.

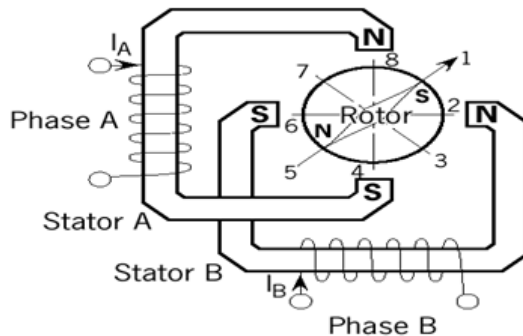
*Motor stepper* disini berfungsi sebagai penggerak *valve* secara elektrik. *Motor stepper* adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. *Motor stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan *motor stepper* diperlukan pengendali *motor stepper* yang membangkitkan pulsa-pulsa periodik (Syahrul, 2005).

Berdasarkan metode perancangan rangkain pengendalinya, *motor stepper* dapat dibagi menjadi jenis *unipolar* dan *bipolar*. Rangkaian pengendali *motor stepper* unipolar lebih mudah dirancang karena hanya memerlukan satu saklar / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (*ground*) pada salah satu terminal lilitan (*wound*) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (VM) pada bagian tengah (*center tap*) dari lilitan (Mukromin, 2017).



**Gambar 2. 2** Motor stepper dengan lilitan unipolar  
(Sumber : <http://zoniaelektro.net/motor-stepper/>)

Untuk *motor stepper* dengan lilitan *bipolar*, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor *unipolar*. *Motor stepper bipolar* memiliki keunggulan dibandingkan dengan *motor stepper unipolar* dalam hal torsi yang lebih besar untuk ukuran yang sama.



**Gambar 2. 3** Motor stepper dengan lilitan bipolar  
(Sumber : <http://zonaelektro.net/motor-stepper/>)

*Motor stepper* bergerak per langkah, dimana setiap langkah mempunyai derajat pergerakan yang sama tergantung dari resolusi dari motor tersebut. *Motor stepper* yang mempunyai resolusi pergerakan yang kecil, pergerakannya lebih baik dibandingkan resolusi yang besar. Hal ini disebabkan karena pergerakan yang besar akan menghasilkan gerakan yang lebih kasar dibandingkan dengan motor yang memiliki resolusi kecil. *Motor stepper* dapat dikendalikan secara *full step* dan *half step*. Pengendalian secara *half step* lebih baik daripada pengendalian secara *full step* karena dengan pengendalian *half step* pergerakan dari motor lebih halus daripada menggunakan pengendalian dengan pengendalian *full step*. *Half Step* adalah cara ara mengendalikan *motor stepper* sehingga menghasilkan pergerakan motor yang lebih halus. Karena pergerakan rotor dalam *motor*

*stepper* yang bergerak dengan sudut sebesar  $\frac{1}{2}$  derajat dari besar sudut antara 2 buah kutub (*coil*) yang berdekatan, sehingga pergerakan yang dihasilkan lebih halus (Mukromin, 2017).

**Tabel 2.1** Pergerakan Half Step

Step	Coil 4	Coil 3	Coil 2	Coil 1	Pergerakan Motor
1	1	0	0	0	
2	1	1	0	0	
3	0	1	0	0	
4	0	1	1	0	

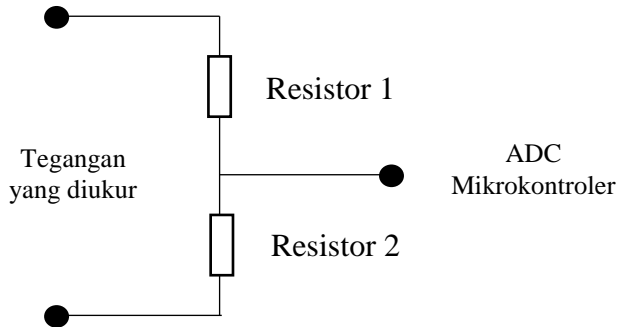
Full step adalah cara mengendalikan motor stepper sehingga dihasilkan pergerakan motor namun tidak sehalus pergerakan half step. Hal ini disebabkan karena pergerakan rotor dalam motor stepper yang bergerak per 1 buah kutub (*coil*). Untuk lebih jelas, konfigurasi motor stepper dengan pengendalian full step untuk setiap pergerakan motor adalah sebagai berikut :

**Tabel 2.2** Pergerakan Full Step

Step	Coil 4	Coil 3	Coil 2	Coil 1	Pergerakan Motor
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	

## 2.7 Prinsip Kerja Sensor Tegangan DC

Prinsip kerja dari sensor tegangan DC adalah dengan membuat perbandingan antara tegangan input dengan tegangan yang akan terbaca dengan mikrokontroler. Pada dasarnya sensor tegangan DC ini terdiri atas rangkaian pembagi tegangan seperti yang terdapat pada gambar dibawah ini.



**Gambar 2. 4** Rangkaian Pembagi Tegangan

Sensor tegangan berfungsi untuk membaca nilai tegangan dari suatu rangkaian. Pada mikrokontroler hanya dapat membaca nilai tegangan sebesar 0-38 volt dengan memanfaatkan pin analog sedangkan jika tegangan yang dibaca lebih dari 38 volt maka memerlukan rangkaian tambahan yakni pembagi tegangan dengan cara mengkonversi nilai yang diterima berupa nilai ADC menjadi nilai tegangan menggunakan rumus sebagai berikut (Fitriandi, Komalasari and Gusmedi, 2016) :

$$V = 106 \times \frac{ADC}{255} \dots \dots \dots (2.6)$$

Dimana :

V = Tegangan (volt)

ADC = Nilai ADC yang terbaca mikrokontroler

255 = Nilai ADC maksimal (8 bit)

106 v = Tegangan referensi ADC mikrokontroler

*(Halaman sengaja Dikosongkan)*

### **BAB III**

## **PERANCANGAN DAN PEMBUATAN ALAT**

### **3.1. Bahan dan Peralatan yang Digunakan**

Pada tugas akhir kali ini membuat sistem pengendalian aliran nosel pada Pembangkit Listrik Tenaga Mikrohidro. Terdapat beberapa alat dan peralatan yang penting didalam rancang bangun alat ini yaitu sebagai berikut :

#### **1. Pompa Air**

Pompa air ini memiliki spesifikasi head minimum 5m dengan kapasitas debit 50 L/menit.



**Gambar 3. 1** Pompa Air

#### **2. Motor Valve**

Pada sistem ini terdapat 1 buah Motor Valve yang terletak diantara pompa dan flow meter yang memiliki spesifikasi 38 volt. Motor Valve ini berfungsi untuk mengatur bukaan/katup yang dialiri air.



**Gambar 3. 2** Motor Valve



### 3. *Voltage Sensor*

Sensor tegangan pada bagian sistem ini digunakan sebagai pendeteksi tegangan yang dikeluarkan oleh generator . Sensor ini memiliki spesifikasi range pengukuran 0-100 volt DC.



**Gambar 3. 3** *Voltage Sensor*

### 4. *Water Flow Sensor*

*Water flow sensor* pada bagian sistem ini digunakan sebagai pendeteksi aliran yang mealui motor valve. Sensor ini memiliki spesifikasi range pengukuran 0-60 LPM.



**Gambar 3. 4** *Water Flow Sensor FS400A G 1"*

### 5. *Mikrokontroler AVR Atmega 128*

*Mikrokontroler AVR Atmega128* merupakan sebuah mikrokontroler yang berbasis pada *input* dan *output* yang mudah digunakan dan sebuah sarana untuk pengembangan yang mengimplementasikan bahasa pemrosesan. AVR ini adalah seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART,

*programmable Watchdog Timerr*, dan *mode power saving*. Mempunyai ADC dan PWM internal. AVR juga mempunyai *In-System Programmable Flash on -chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. Atmega128 adalah mikrokontroler CMOS 8 -bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus clock, Atmega128 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi komsumsi daya versus kecepatan proses.



**Gambar 3. 5** Mikrokontroler ATMega 128

#### 6. *Driver Motor*

*Driver motor* yang digunakan adalah L289N sebagai penggerak *motor stepper* yang terhubung dengan *valve*.



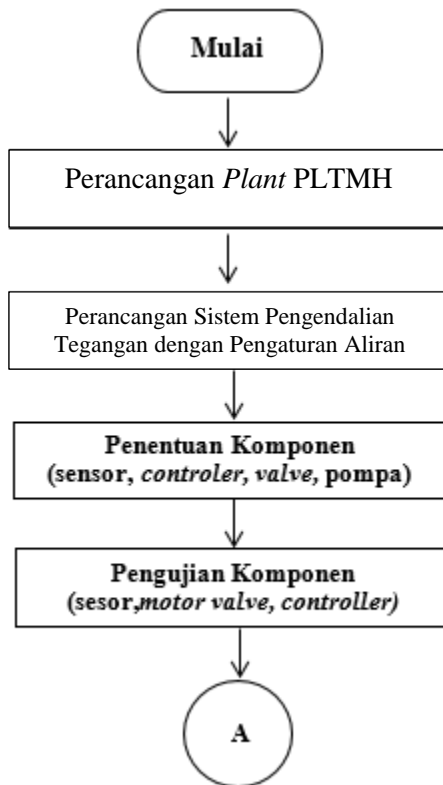
**Gambar 3. 6** *Driver Motor L289N*

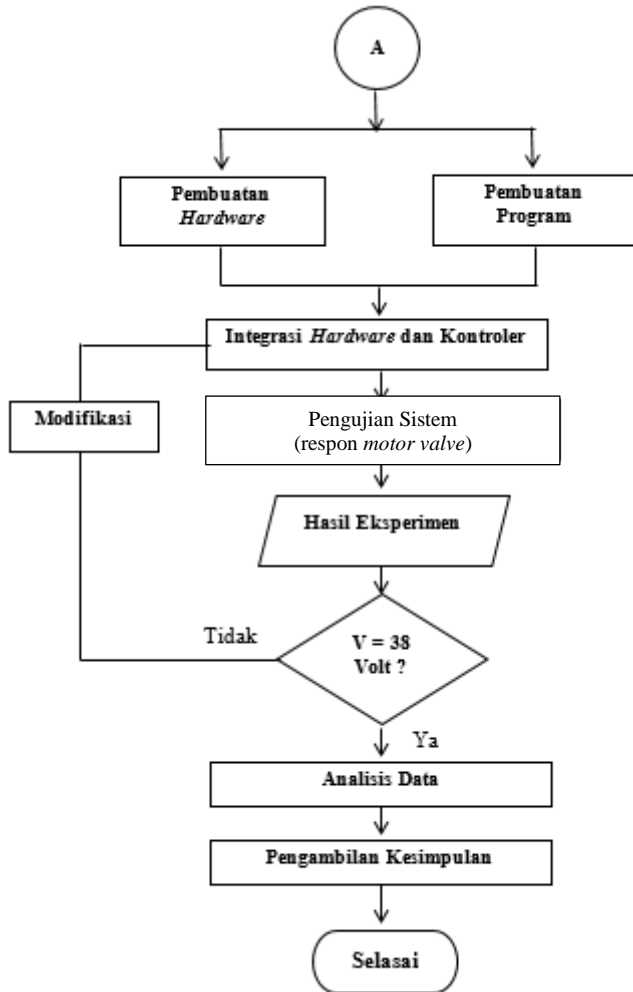
### 3.2. Metodologi Perngerjaan

Metodologi pengerjaan pada Tugas akhir ini adalah sebagai berikut :

#### 3.2.1 *Flowchart* Perancangan Alat

Berikut ini adalah Langkah-langkah perancangan alat ini digambarkan dalam *flowchart* pengerjaan adalah sebagai berikut :





**Gambar 3. 7** Flowchart Pengerjaan Tugas Akhir

1. Perhitungan Dimensi dan Perancangan sistem *Hardware dan Software*.

Pada tahap ini dilakukan perhitungan dimensi serta spesifikasi dari nosel yang dibutuhkan untuk memutar turbin dengan memperhitungkan debit air yang mampu disuplai. Pada tahap ini juga dilakukan perancangan sistem *hardware* kontrol debit yang dibutuhkan dan perancangan perangkat lunak

## 2. Pembuatan Alat

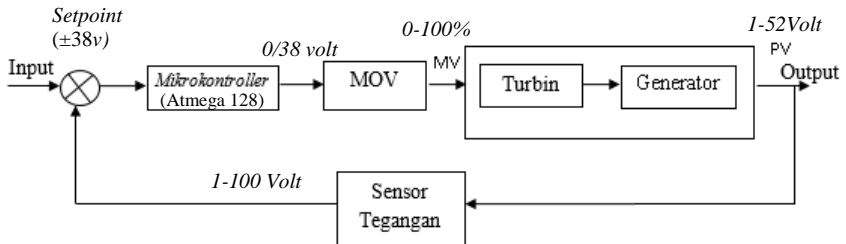
Tahapan ini yaitu dilakukan perakitan atau penggabungan komponen-komponen yang dibutuhkan berupa *sensor*, *controller*, *actuator*, *display*, dan pembuatan program.

## 3. Integrasi *Hardware*, *Software*, dan Pengujian Alat

Pada tahapan ini alat yang sudah dirakit serta program yang telah dibuat diuji dengan dialiri debit yang sesuai untuk mengetahui apakah alat yang sudah dibuat dapat berfungsi sebagaimana yang diinginkan.

## 4. Pengambilan Data Besaran Fisis

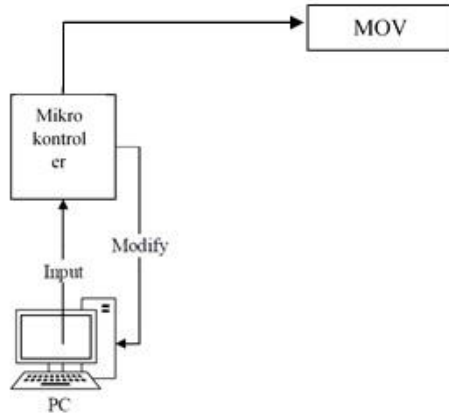
Pengambilan data yang dilakukan dengan menghitung besar tegangan, rpm, dan debit aliran.



**Gambar 3. 8** Diagram Blok Pengendalian

Pada gambar 3.8 menjelaskan mengenai alur sistem pengendalian yang saling terintegrasi untuk mencapai nilai *set point* yang ditentukan. Nilai *set point* yang diberikan akan akan diproses oleh mikrokontroler Atmega128 sebagai kontroler sekaligus akan diproses untuk memerintahkan aktuator. Setelah

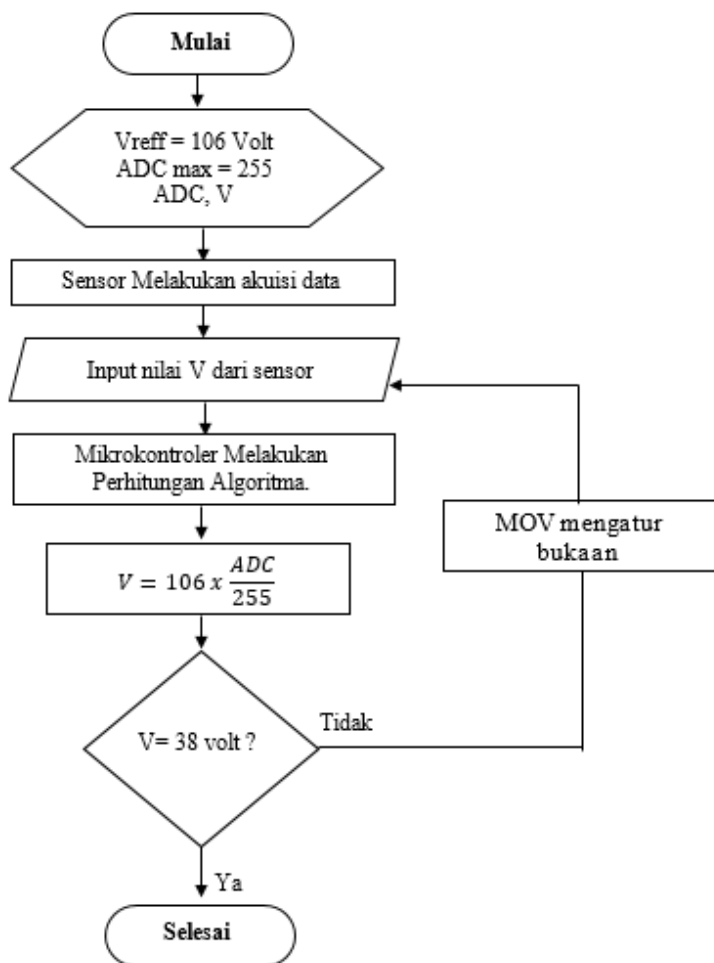
sinyal *input* diproses oleh mikrokontroler, MOV (Motor Valve) akan mengatur besar bukaan sesuai dengan yang dibutuhkan.



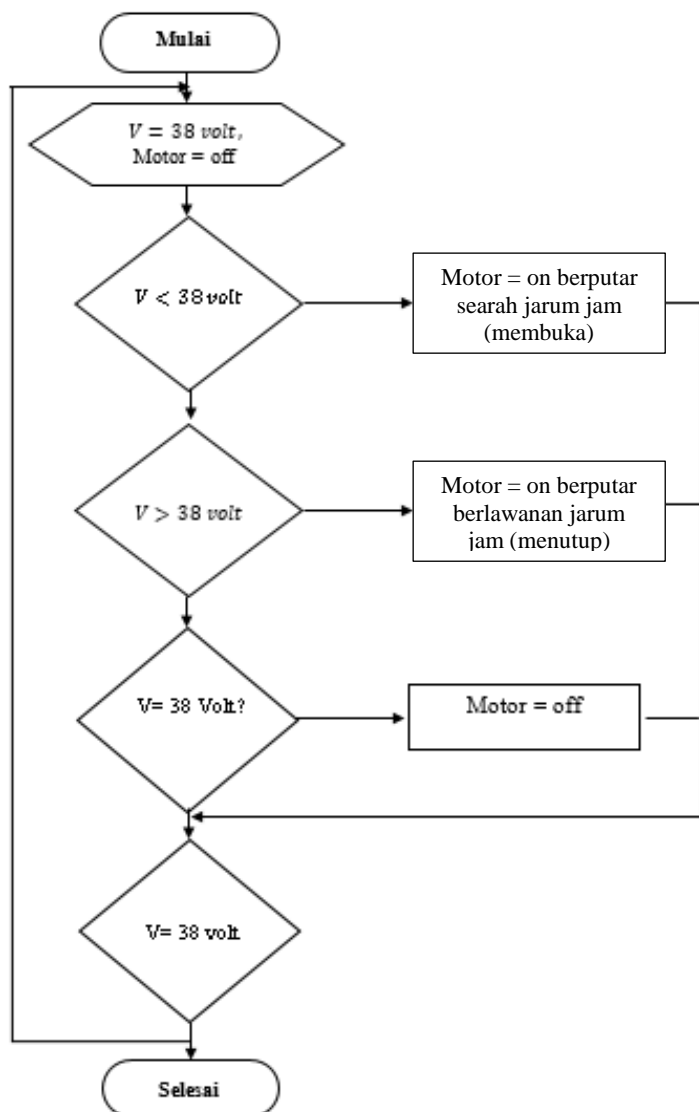
**Gambar 3. 9** Diagram Kerja Sistem  
(Sumber : Wirabudi, 2017)

### 3.3. Perancangan Alat Tugas Akhir

Perancangan Sistem pengendalian tegangan dengan pengaturan laju aliran air dijelaskan sebagai berikut :



**Gambar 3. 10** Diagram Alir Integrasi Sensor, *controller*, dan Aktuator

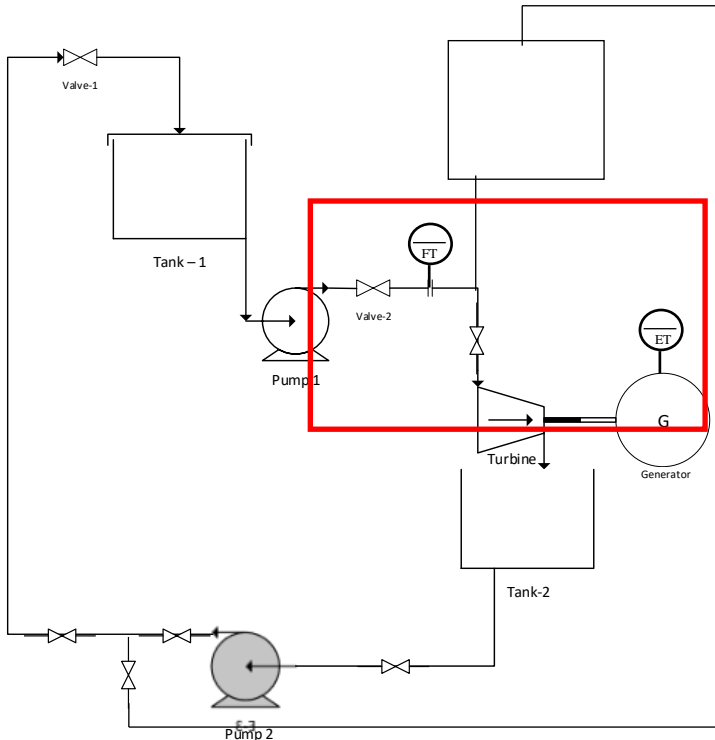


**Gambar 3. 11** Diagram Alir pengendalian Motor Valve



### 3.3.1 Perancangan PFD, dan P&ID

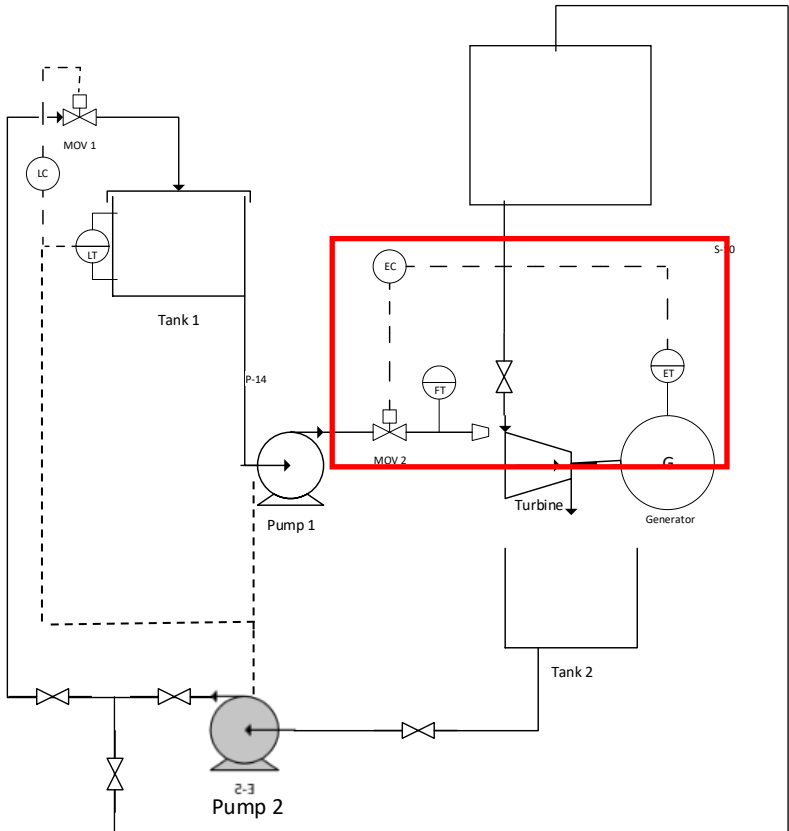
Desain yang dibuat adalah sebagai berikut:



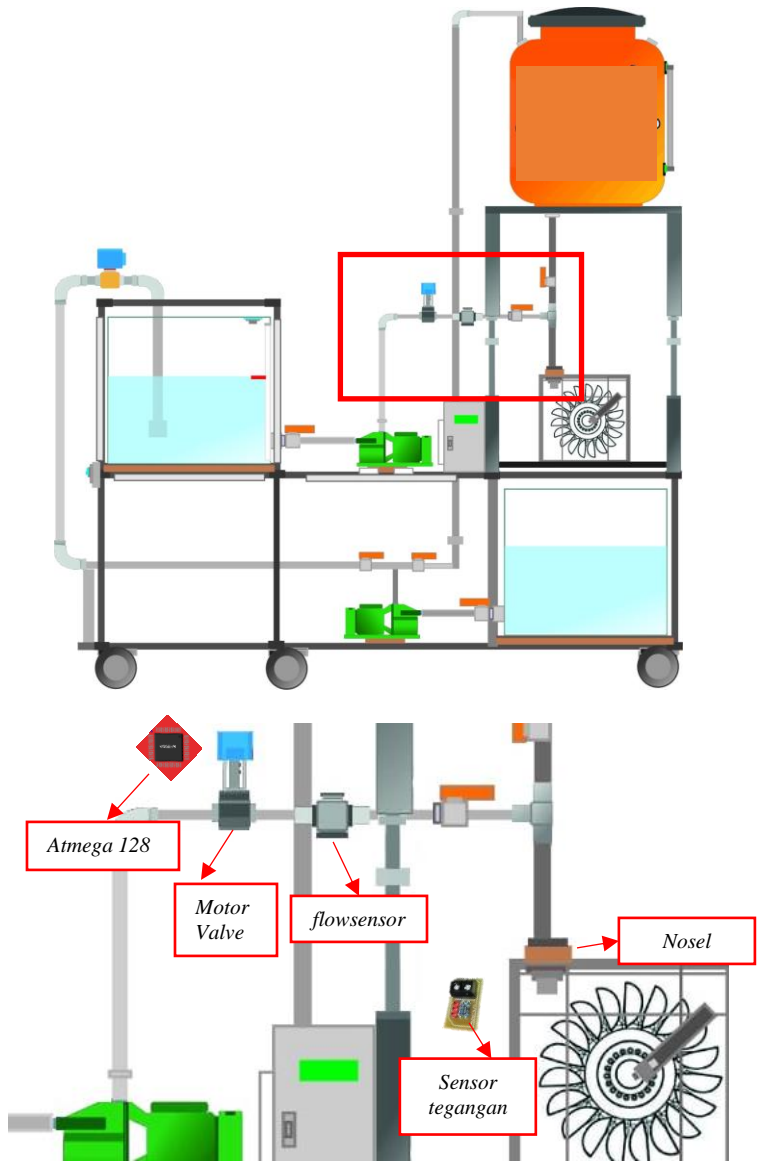
**Gambar 3. 12** *Process Flow Diagram*

Pada plan ini dibuat desain *process flow diagram* seperti pada Gambar 3.9. Pada *process flow diagram* dapat dijelaskan sebuah proses alir dimana pada awalnya sensor flow akan melakukan akuisi data pada aliran air, lalu data yang didapat diteruskan ke mikrokontroler yang akan melakukan perhitungan algoritma dan memerintahkan pada aktuatur untuk mengatur bukaan sesuai dengan debit yang dibutuhkan.

Setelah desain dari *Proses Flow Diagram* dibuat maka langkah selanjutnya yaitu membuat desain *Piping and Instrument Diagram* seperti pada gambar 3.12.



**Gambar 3. 13** *Piping and Instrument Diagram*



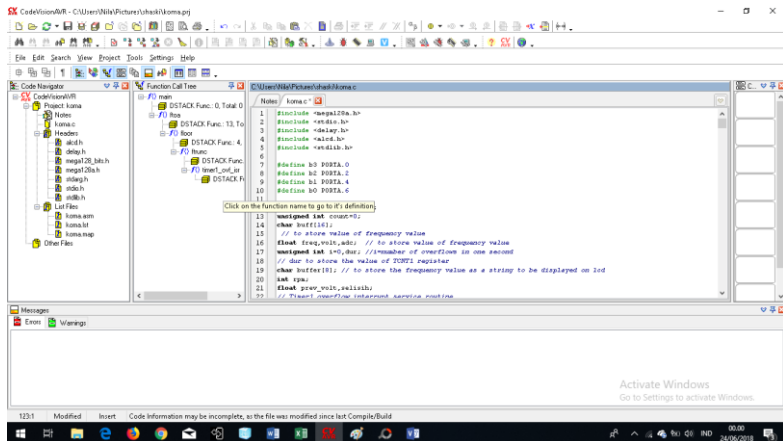
Gambar 3. 14 Design Plant PLTMH

### 3.4 Perancangan Perangkat Lunak (Software)

Pada perancangan alat ini terdapat *software* yang digunakan untuk membuat program sebelum di-*compile* ke mikrokontroler dan digunakan sebagai *compiler* program ke, yaitu *Code Vision AVR V 2.05.0*.

#### 3.4.1 *Code Vision AVR V 2.05.0*

Seperti yang terlihat pada gambar 3.15. *software code vision AVR v 2.05.0* digunakan untuk membuat *listing program* yang berisi perintah - perintah guna mengintegrasikan hasil pengukuran tegangan yang dihasilkan oleh generator oleh sensor menuju ke mikrokontroler atmega128. Selain itu, dapat juga memberikan perintah untuk mengintegrasikan mikrokontroler menuju ke *liquid crystal display 20x4* serta *personal computer*.

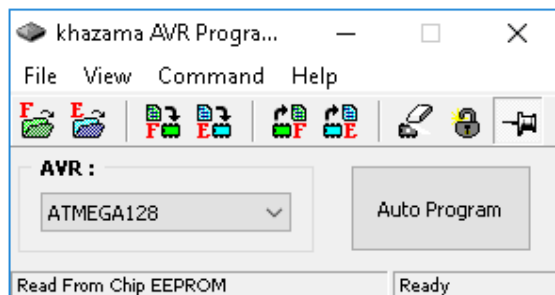


Gambar 3. 15 Tampilan Software Code Vision AVR V 2.05.0

#### 3.4.2 Khazama V 1.7.0

*Software* ini digunakan untuk meng-*compile listing* program yang dibuat pada *code vision AVR* ke mikrokontroler atmega128. Selain itu, khazama v 1.7.0 juga dapat melakukan pengaturan *clock* dan proses eksekusi program pada mikrokontroler. Tampilan

khazama v 1.7.0 yang digunakan pada tugas akhir kali ini dapat dilihat pada gambar 3.16.



**Gambar 3. 16** Khazama V 1.7.0

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Rancang Bangun Alat**

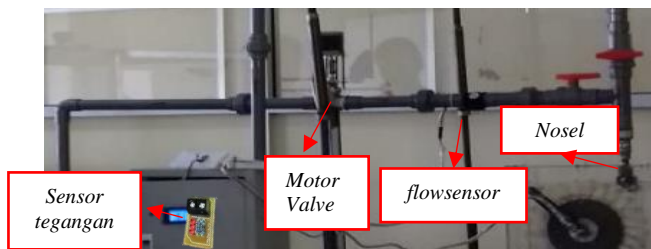
Berikut ini adalah rancang bangun Pembangkit Listrik Tenaga Mikrohidro.



**Gambar 4. 1** Rancang Bangun PLTMH

##### **4.1.1 Perancangan Sistem Pengendalian Tegangan**

Berikut ini adalah perancangan sistem pengendalian tegangan dengan pengaturan laju aliran nosel pada PLTMH berbasis ATmega128.



**Gambar 4. 2** Sistem Pengendalian tegangan dan laju aliran.

Rancang bangun PLTMH ini menggunakan fluida air sebagai tenaga penggeraknya untuk memutar turbin, dengan cara memanfaatkan beda ketinggian dan jumlah debit air yang dipicu dengan pompa dan akan memutar Turbin. Dari banyaknya putaran yang dihasilkan oleh turbin berdasarkan jumlah debit yang mengalir, maka generator akan mengubah putaran tersebut menjadi tegangan. Agar didapatkan tingkat efektifitas dan efisiensi debit aliran air yang lebih tinggi untuk menghasilkan jumlah putaran turbin dan tegangan yang stabil.



**Gambar 4. 3** Lampu LED

Proses pengendalian tegangan ini menggunakan sensor tegangan dengan aktuator berupa *motor valve* yang dapat mengendalikan laju aliran nosel pada PLTMH yang keluarannya digunakan untuk menghidupkan lampu 3 buah lampu yang masing-masing membutuhkan tegangan sebesar 12 volt maka setpoint yang diperlukan adalah tegangan sebesar 38 Volt.

#### **4.1.2 Pengujian Komponen**

Pengujian komponen dilakukan secara terpisah antara pompa, sensor tegangan, motor valve, mikrokontroler.

##### **4.1.2.1 Pengujian Pompa Air**

Pengujian Pompa air dilakukan untuk melihat nilai debit aliran air yang dipompa dari tangki. Proses pengambilan data dilakukan dengan cara memompa air dari tangki atas yang melalui *motor valve* dengan kondisi *Full Open* dan *water flow sensor*

selama 50 detik. *Water flow sensor* mendeteksi debit aliran yang melaluinya dan hasil terekam secara otomatis pada *sdcard*.  
Dibawah ini adalah hasil pengujian pompa sebagai berikut :

**Tabel 4. 1** Hasil Pengujian Pompa

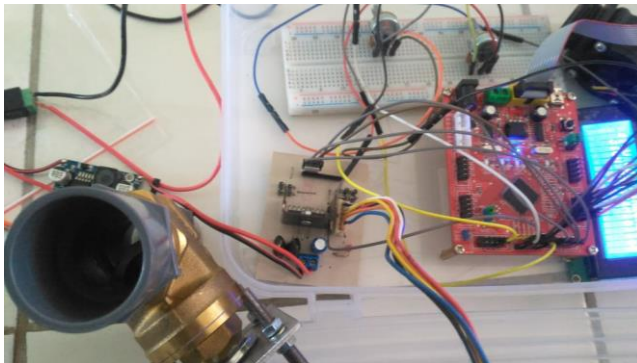
Waktu (detik)	Debit (LPM)	Waktu (detik)	Debit (LPM)
1	26,75	26	28,87
2	26,75	27	28,87
3	26,75	28	26,75
4	26,75	29	26,75
5	26,75	30	28,87
6	28,87	31	28,87
7	28,87	32	28,87
8	28,87	33	28,87
9	28,87	34	28,87
10	28,87	35	28,87
11	28,87	36	26,75
12	28,87	37	26,75
13	26,75	38	26,75
14	26,75	39	28,87
15	26,75	40	28,87
16	28,87	41	28,87
17	28,87	42	26,75
18	26,75	43	26,75
19	26,75	44	28,87
20	28,87	45	28,87
21	28,87	46	28,87
22	28,87	47	28,87
23	26,75	48	26,75
24	26,75	49	26,75
25	28,87	50	26,75
<b>Rata-rata Debit</b>		<b>27,9372 LPM</b>	



Berdasarkan tabel 4.1 hasil pengujian pompa air dengan bukaan maksimal diperoleh hasil rata-rata debit aliran air yang dipompa dari tangki atas menuju turbin sebesar 27,9372 LPM. Maka dapat disimpulkan bahwa debit keluaran pompa tidak stabil yang disebabkan karena terjadi perubahan daya listrik yang dikonsumsi.

#### 4.1.2.2 Pengujian *Motor Valve*

Pengujian *motor valve* dilakukan untuk melihat perbedaan tegangan yang dihasilkan oleh generator karena dipengaruhi oleh besar bukaan oleh *motor valve*. Rangkaian *Motor valve* disambungkan dengan *driver motor* dan *power supply 5V/5A* untuk menggerakkan *motor valve*. Dibawah ini adalah gambar rangkaian motor valve yang digunakan pada sistem ini.



**Gambar 4. 4** Rangkaian *motor valve*

Pengujian *motor valve* ini dilakukan dengan cara membandingkan tegangan yang dihasilkan oleh generator terhadap bukaan *valve* yang berbeda-beda. *Valve* diatur dengan bukaan sebesar 50% ( $1.080^\circ$ ), 73% ( $1.577^\circ$ ), dan 100% ( $2.160^\circ$ ). Berikut ini adalah hasil pengujian bukaan *motor valve* terhadap tegangan yang dihasilkan generator.

**Tabel 4. 2** Hasil Pengujian *Motor Valve*

Waktu (Detik)	Tegangan (volt)		
	Bukaan Valve		
	50%	73%	100%
1	0,42	4,35	6,57
2	2,95	16,25	17,7
3	6,95	18,38	24,63
4	15,24	21,58	30,02
5	19,52	22,78	34,35
6	22,32	26,8	38,48
7	23,1	25,96	40,59
8	24,55	31,71	44,34
9	25,32	34,72	45,38
10	27,72	34,9	46,56
11	29,71	36,12	47,97
12	31,65	36,29	48,28
13	33,47	37,88	50,51
14	32,24	39,99	50,94
15	32,4	40,12	50,07
16	33,21	41,55	49,19
17	33,36	41,07	51,14
18	33,27	42,31	51,05
19	32,09	41,19	50,75
20	30,02	39,95	51,42
21	31,72	37,75	51,49
22	30,02	37,07	51,37
23	29,81	36,12	51,24
24	29,28	36,29	50,77
25	31,72	37,88	50,89
26	33,27	37,36	51,6
27	32,09	37,32	52,05
28	30,02	39,73	50,43
29	29,59	40,21	52,05
<b>Rata-Rata Tegangan</b>	<b>26,55</b>	<b>33,79</b>	<b>44,80</b>

Berdasarkan tabel 2.1 hasil pengujian *motor valve* dapat diketahui bahwa dengan adanya perubahan besar bukaan *valve* akan mempengaruhi tegangan yang dihasilkan generator. Pada tabel diatas didapatkan nilai rata-rata tegangan yang dihasilkan generator dengan bukaan 50%,73%, dan 100% masing-masing sebesar 26,55 volt, 33,79 volt, dan 44,80 volt.

#### 4.1.2.3 Pengujian Sensor Tegangan

Sensor tegangan yang digunakan, harus mampu membaca tegangan yang dihasilkan oleh generator dengan sesuai, untuk itu dilihat dari karakteristik dari sensor tegangan ini.

Pengujian sensor tegangan ini dilakukan dengan menghubungkan sensor tegangan dengan mikrokontroler dan diberikan sumber tegangan yang akan dibaca oleh sensor dan ditampilkan melalui LCD. Hasil pembacaan oleh sensor dibandingkan dengan hasil pembacaan pada alat ukur multimeter. Pengukuran dilakukan dengan diberikan 5 range pengukuran (2, 4, 6, 8, dan 10) secara naik dan dilanjutkan dengan pengambilan data secara turun.

Berikut ini adalah perbandingan hasil pengukuran tegangan oleh sensor dan multimeter.

**Tabel 4. 3** Hasil Pengambilan Data Naik

No	Pembacaan standar	Pembacaan Alat					Rata-Rata
		Data Ke					
		1	2	3	4	5	
1	2	1,99	2,04	1,97	1,97	2,02	1,998
2	4	4,01	3,98	4,03	4,01	4,03	4,012
3	6	6,07	6,05	6,09	6,19	6,07	6,094
4	8	8,17	8,11	8,13	8,16	8,13	8,14
5	10	10,08	10,1	10,08	10,1	10,05	10,082

**Tabel 4. 4** Hasil Pengambilan Data Turun

No	Pembacaan standar	Pembacaan Alat					Rata-Rata
		Data Ke					
		1	2	3	4	5	
1	2	2,03	2,05	2,05	2,03	2,03	2,038
2	4	4,06	4,09	4,04	4,06	4,06	4,062
3	6	6,15	6,13	6,15	6,13	6,17	6,146
4	8	8,17	8,14	8,19	8,17	8,19	8,172
5	10	10,27	10,28	10,24	10,27	10,29	10,27

Berdasarkan data diatas, maka diperoleh rata-rata setiap data naik dan turun, untuk diolah lagi agar didapatkan nilai koreksi atau error antara data pembacaan alat ukur standar (multimeter) dengan pembacaan oleh sensor .

Dibawah ini adalah data kalibrasi sensor tegangan sebagai berikut :

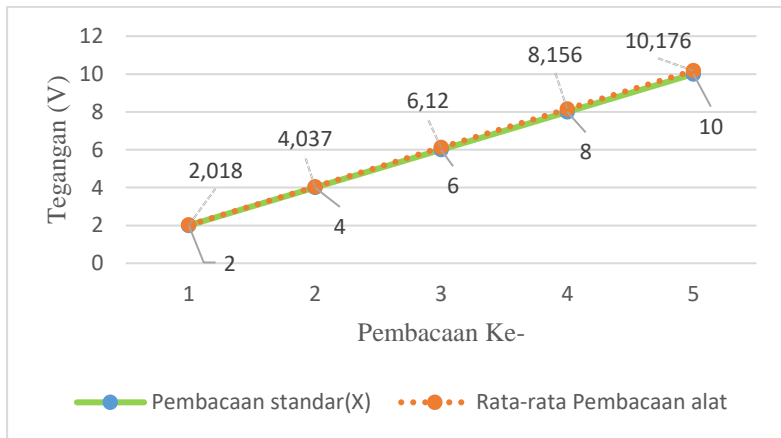
**Tabel 4. 5** Data Kalibrasi Sensor

No	Pembacaan standar(X)	Pembacaan Alat		Rata-rata	Koreksi (Y)
		Rata-rata Naik	Rata-rata Turun		
1	2	1,998	2,038	2,018	-0,018
2	4	4,012	4,062	4,037	-0,037
3	6	6,094	6,146	6,12	-0,12
4	8	8,14	8,172	8,156	-0,156
5	10	10,082	10,27	10,176	-0,176
<b>Jumlah</b>					<b>-0,507</b>
<b>Rata-Rata</b>					<b>-0,1014</b>

**Tabel 4. 6** Data Kalibrasi Lanjutan

No	X	X <sup>2</sup>	Y-Y'	X*Y	(Y-Y') <sup>2</sup>
1	2	4	0,0834	-0,036	0,00695556
2	4	16	0,0644	-0,148	0,00414736
3	6	36	-0,0186	-0,72	0,00034596
4	8	64	-0,0546	-1,248	0,00298116
5	10	100	-0,0746	-1,76	0,00556516
<b>Jumlah</b>	<b>30</b>	<b>220</b>	<b>0</b>	<b>-3,912</b>	<b>0,0199952</b>
<b>Rata-Rata</b>	<b>6</b>	<b>44</b>	<b>0</b>	<b>-0,7824</b>	<b>0,00399904</b>

Berdasarkan Tabel 4.4 Data kalibrasi sensor didapatkan grafik perbandingan pembacaan rata-rata sensor dan alat ukur multimeter. Berikut ini adalah grafik perbandingan antara pembacaan oleh sensor dan multimeter.

**Gambar 4. 5** Grafik Pembacaan oleh Sensor Tegangan dan Multimeter

Berikut ini merupakan cara untuk mencari nilai ketidakpastian sensor Tegangan.

a. Nilai Ketidakpastian Tipe A

$$\sigma = \frac{\sqrt{\sum(y-\bar{y})^2}}{n-1} \dots\dots\dots(4.1)$$

$$\sigma = \frac{\sqrt{0,0199}}{5-1} = 0,070$$

Sehingga didapatkan nilai  $U_{a1}$  sebagai berikut:

$$U_{a1} = \frac{\sigma}{\sqrt{n}} \dots\dots\dots(4.2)$$

$$U_{a1} = \frac{0,070}{\sqrt{5}} = 0,0316$$

$$\text{Sedangkan nilai ketidakpastian regresi } U_{a2} = \sqrt{\frac{SSR}{n-2}}$$

Dimana:

$SSR$  (*Sum Square Residual*)=  $\sum SR$  (*Square Residual*)

$SR = R^2$  (*Residu*)

$Y$  = Nilai koreksi

$$Y_{reg} = a + (b \cdot x_i) \dots\dots\dots(4.3)$$

$$a = \bar{y}_i + (b \cdot \bar{x}) \dots\dots\dots(4.4)$$

$$b = \frac{n \cdot \sum xy - \sum x \sum y}{n \cdot \sum x^2 - (\sum x)^2} \dots\dots\dots(4.5)$$

$$SSR = \sum (y - Y_{reg})^2 \dots\dots\dots(4.6)$$

Dimana:

$x$  = *Pemb. standar*,

$y$  = *Nilai koreksi*,  $n$  = *Jumlah data*

**$b$**  = 0,0291

Sehingga nilai:

$$\mathbf{a} = -0,02175$$

Jadi, persamaan regresi menjadi:

$$Y_{reg} = (-0,02175) + (x(0,0291))$$

Yang menghasilkan nilai **SSR = 0,001**

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} = \sqrt{\frac{0,001}{5-2}} = 0,018$$

- b. Nilai Ketidakpastian *Type B* :

Pada tipe ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi ( $U_{b1}$ ) dan Ketidakpastian alat standar *pressure gauge* ( $U_{b2}$ ). Dengan perhitungan sebagai berikut:

$$U_{b1} = \frac{\frac{1}{2}x_{Resolusi}}{\sqrt{3}} = \frac{\frac{1}{2}x_1}{\sqrt{3}} = \mathbf{0,2886751}$$

$U_{b2} = \frac{a}{k}$ , dikarenakan pada alat standar tidak ada sertifikat kalibrasinya maka nilai  $a$  (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0. Sehingga hasil :  $U_{b2} = 0$

- c. Nilai ketidakpastian kombinasi  $U_c$ :

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2} \dots\dots\dots(4.7)$$

$$U_c = \sqrt{0,0316^2 + 0,019^2 + 0,2886^2 + 0^2}$$

$$U_c = 2,07696$$

## 4.2 Pengujian Sistem

Sensor tegangan pengujian sensor dilakukan dengan memberikan tegangan menggunakan *power supply*. Hasil

pembacaan sensor dapat dilihat seperti pada tabel 4.5. pengambilan data dilakukan setiap 2 detik. Actuator motor valve dapat memberikan output mendekati setpoint sensor sebesar 38 volt. Pengujian dilakukan untuk mengetahui grafik respon actuator dan pembacaan tegangan pada sensor berdasarkan waktu yang telah ditentukan.

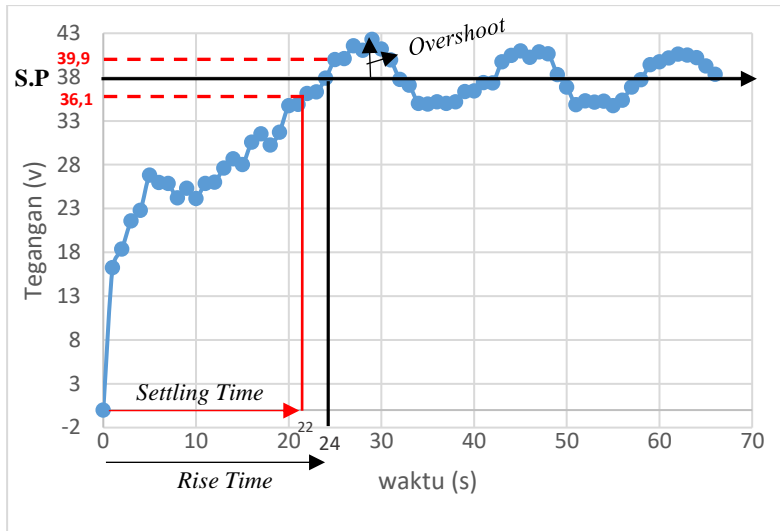
**Tabel 4. 7** Hasil Pengujian Sistem

Waktu (s)	Tegangan (v)	Flow (LPM)	Waktu (s)	Tegangan (v)	Flow (LPM)
0	0	0	16	30,57	22,75
1	16,25	16,62	17	31,53	22,75
2	18,38	16,62	18	30,26	24,75
3	21,58	16,62	19	31,71	24,75
4	22,78	18,62	20	34,72	22,75
5	26,8	18,62	21	34,9	24,75
6	25,96	20,5	22	36,12	22,75
7	25,87	22,75	23	36,29	24,75
8	24,2	22,75	24	37,88	24,75
9	25,32	20,5	25	39,99	22,75
10	24,15	20,5	26	40,12	24,75
11	25,86	20,5	27	41,55	24,75
12	25,99	22,75	28	41,07	24,75
13	27,62	22,75	29	42,31	24,75
14	28,66	22,75	30	41,19	20,5
15	28,01	22,75	31	39,95	22,75



Waktu (s)	Tegangan (v)	Flow (LPM)	Waktu (s)	Tegangan (v)	Flow (LPM)
32	37,75	20,5	50	36,87	20,5
33	37,07	20,5	51	34,86	20,5
34	35,01	22,75	52	35,29	22,75
35	34,93	22,75	53	35,12	22,75
36	35,17	20,5	54	35,23	24,75
37	34,98	24,75	55	34,74	24,75
38	35,17	24,75	56	35,36	22,75
39	36,36	22,75	57	36,87	24,75
40	36,43	24,75	58	37,69	24,75
41	37,36	24,75	59	39,43	24,75
42	37,32	24,75	60	39,73	22,75
43	39,73	24,75	61	40,18	24,75
44	40,47	24,75	62	40,6	24,75
45	41	24,75	63	40,52	22,75
46	40,26	22,75	64	40,22	22,75
47	40,86	22,75	65	39,27	22,75
48	40,67	22,75	66	38,3	20,5
49	38,28	22,75			

Berdasarkan data tersebut maka diperoleh sebuah grafik pengujian sensor sebagai berikut :



**Gambar 4. 6** Grafik Uji Sistem

Grafik 4.5 menunjukkan hasil Respon uji sistem pengendalian tegangan terhadap waktu dengan *setpoint* 38 volt didapatkan nilai sebagai berikut :

1. *Rise time*, ukuran waktu yang diukur mulai dari respon detik ke 0 sampai dengan respon memotong *steady state* pertama selama 24 detik.
2. *Maximum Overshoot*, nilai maksimum dari kurva respons yang diukur. Nilai *peak time* pada sistem didapatkan sebesar 42,31 Volt.

$$Mp = \frac{C_{(tp)} - C_{\infty}}{C_{\infty}} \times 100\% \dots \dots \dots (4.8)$$

Dimana :

$Mp$  = *Maximum Overshoot*

$C_{(tp)}$  = Nilai *peak time* (Nilai Tertinggi)

$C_{(\infty)}$  = Nilai *setpoint*

$$Mp = \frac{42,31 - 38}{38} \times 100\% = 11,34\%$$

3. *Settling time*, waktu yang dibutuhkan untuk mencapai *steady state* dengan persentase  $\pm 5\%$  dari *setpoint*. *Settling time* pada pengendalian tegangan dengan *sepoint* 38 volt didapatkan nilai sebagai berikut.

$$t_c = S.P \pm (S.P \times 5\%) \dots\dots\dots(4.9)$$

Dimana :

$t_c$  = *Settling Time*

$S.P$  = *Setpoint*

$$t_c = 38 \pm (38 \times 5\%)$$

$$t_c = 38 \pm 1,9$$

$$t_c = 39,9 \text{ volt selama 25 detik atau}$$

$$t_c = 36,1 \text{ volt selama 22 detik}$$

Gambar 4.6 menunjukkan bahwa nilai *rise time* pada uji respon sistem pengendalian tegangan selama 22 detik.

Berdasarkan grafik respon pengendalian diatas dapat diketahui bahwa hasil respon sistem tegangan oleh *motor valve* terhadap waktu kurang stabil dan berosilasi karena menggunakan pengendalian *if codition* atau *fuzzy*.

### 4.3 Pembahasan

Dilakukan pengujian pengendalian tegangan dengan pengaturan laju aliran menggunakan aktuator motor valve bertujuan untuk mengatur jumlah putaran turbin agar tidak terjadi *overload* pada generator. Motor valve diletakkan pada input sensor flow sebelum nosel, sedangkan sensor tegangan dipasang pada output generator. Sensor tegangan yang digunakan dengan prinsip pembagi tegangan dan range pengukuran antara 1-100 Volt. Rangkaian driver motor dc menggunakan rangkaian modul driver L298 yang mampu bekerja hingga arus 4A, sedangkan beban motor yang digunakan yaitu 3A. Berdasarkan kalibrasi sensor

tegangan didapatkan ketidakpastian  $U_{a1}$  sebesar 0,031. Pada uji respon sistem motor valve yang terdapat pada tabel 4.5 didapatkan nilai *rise time* dan *settling time* selama 24 detik dan nilai *overshoot* sebesar 42,31 volt dan pada gambar 4.5 grafik uji sistem menunjukkan beresilasi karena menggunakan mode pengendalian *if condition* atau *fuzzy*. Pengendalian tegangan yang dilakukan dengan pengaturan laju aliran melalui bukaan motor valve agar sesuai dengan set point yang ditentukan. Pada percobaan kali menghidupkan 3 buah lampu yang membutuhkan tegangan sebesar 38 volt, maka motor valve akan membuka atau menutup hingga sensor menunjukkan pengukuran sebesar 38 volt, pada pengendalian ini menggunakan 3 kondisi jika pembacaan tegangan kurang dari 38 maka motor valve akan bergerak membuka dan sebaliknya hingga mencapai set point. Pada grafik respon sistem didapatkan nilai *rise time* sebesar 24 detik, nilai *maximum overshoot* sebesar 11,24%, dan waktu yang dibutuhkan untuk mencapai *steady state* adalah 22 selama detik.

*(Halaman Sengaja Dikosongkan)*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan rancang bangun dan analisa data yang telah dilakukan maka dapat disimpulkan bahwa :

- Terjadinya *osilasi* pada pembacaan tegangan pada generator dengan *setpoint* yang ditentukan yaitu 38 volt menggunakan sensor tegangan dan aktuator berupa *motor stepper*.
- Aktuator berupa motor stepper berjalan sesuai *setpoint* yang diinginkan yaitu saat pembacaan tegangan mendekati 38 volt maka *motor stepper* akan berhenti dan valve akan diam. Data pengujian alat pada sistem pengendalian , didapatkan *respon time* yang cukup baik untuk mencapai *steady state* yaitu nilai *rise time* sebesar 24 detik, *maximum overshoot* sebesar 11,24%, dan *settling time* yang dibutuhkan untuk mencapai *steady state* selama 22 detik.

#### **5.2 Saran**

Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan adalah memastikan bahwa plant sudah tidak ada yang bocor untuk menghindari *losse* saat pengendalian karena akan mempengaruhi laju aliran dan membuat pengukuran tegangan menjadi tidak stabil.

*(Halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

Abdur Rohman , M. Agung Prawira Negara, B. S. (2017) ‘Sistem Pengaturan Laju Aliran Air pada Plant Water Treatment Skala Rumah Tangga dengan Kontrol Fuzzy-Pid’, 1, pp. 29–34.

Ceri Steward Poeta, D. (2013) ‘Pembangkit Listrik Di Desa Kali Kecamatan Pineleng Dengan Head 12 Meter’.

Dr. Aris Triwiyatno, S. M. (2008) ‘Bab 9 dasar sistem kontrol’, pp. 169–283.

Firmansyah, M. (2013) ‘Distribusi Tekanan Fluida pada Nozel Turbin Pelton Berskala Mikro dengan Menggunakan Perangkat Lunak Solidworks’.

Fitriandi, A., Komalasari, E. and Gusmedi, H. (2016) ‘Rancang Bangun Alat Monitoring Arus dan Tegangan Berbasis Mikrokontroler dengan SMS Gateway’, *Jurnal Rekayasa dan Teknologi Elektro*, 10(2), pp. 87–98. doi: 2549-3442.

Khalid, A. (2016) ‘Rancang Bangun Miniatur Pembangkit Listrik Tenaga Air ( PLTA )’, 16(1), pp. 22–30.

Maulana, F. A. and Dkk (2017) ‘Prototipe Kontrol Aliran Air Berbasis Prototype of Water Flow Control Based on’, 4(2), pp. 1434–1440.

Mukromin, R. indra (2017) *RANCANG BANGUN SISTEM PENGENDALIAN FLOW RATE INLET KOLOM PURIFIKASI PADA PROSES PURIFIKASI BIOGAS DENGAN WATER SCRUBBER SYSTEM BERBASIS ATMEGA 128*.

Pasalli, Y. R. and Rehiara, A. B. (2014) ‘Design Planning of Micro-hydro Power Plant in Hink River’, *Procedia Environmental Sciences*. Elsevier B.V., 20, pp. 55–63.



Sofyana, N. U. (2017) 'RANCANG BANGUN SISTEM PENGENDALIAN LAJU ALIRAN DAN REKONSILIASI DATA PADA SPLITTING PIPING'.

Sukamta, S. and Kusmantoro, A. (2015) 'Perencanaan pembangkit listrik tenaga mikro hidro (PLTMH) Jantur Tabalas Kalimantan Timur', *Jurnal Teknik Elektro*, 5(2), pp. 58–63.

Syahrul (2005) 'Motor Stepper: Teknologi, Metoda dan Rangkaian Kontrol', *Majalah Ilmiah UNIKOM*, 6(2), pp. 187–202.

Trisasiwi, W. (2017) 'RANCANG BANGUN TURBIN CROSS-FLOW UNTUK PEMBANGKIT LISTRIK TENAGA MIKROHIDRO (PLTMH) SKALA LABORATORIUM', 13(1), pp. 29–36.

Winanti, N. (2014) 'PROTIPE PLTA DENGAN MEMANFAATKAN ENERGI KINETIK AIR UNTUK PENERANGAN', *Prosiding Seminar Nasional Sains dan Pendidikan Sains IX, Fakultas Sains dan Matematika, UKSW*, 5.

Wirabudi, W. I. (2017) 'PENGEMBANGAN SISTEM KONTROL GATE PADA FLOW CONTROL VALVE SISTEM PEMBANGKIT DAYA MIKROHIDRO BERBASIS PLC SIEMENS S7-1200'.



**LAMPIRAN A**  
**(LISTING PROGRAM PADA CODE VISION AVR)**

```
/******  
****
```

This program was produced by the  
CodeWizardAVR V2.05.3 Standard  
Automatic Program Generator  
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.  
<http://www.hpinfotech.com>

Project : PLTMH TEAM  
Version :  
Date : 07/07/2018  
Author : PLTMH  
Company : ITS  
Comments:

Chip type : ATmega128A  
Program type : Application  
AVR Core Clock frequency: 16,000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 1024

```
*****  
***/
```

```
#include <mega128a.h>  
#include <alcd.h>  
#include <stdlib.h>  
#include <delay.h>  
#include <stdio.h>  
#include <i2c.h>  
#include <ds1307.h>
```

```

#define pompa1 PORTD.4
#define pompa2 PORTD.5
#define mov1 PORTB.6
#define mov2 PORTB.7
#define b3 PORTA.0
#define b2 PORTA.2
#define b1 PORTA.4
#define b0 PORTA.6

char temp1 [16];
int waktu = 0,persen;
unsigned char Rpm, Rpm2;
float jrk,acuan_mov; //,acuan_mov_min;
int jarak,count=0,drajat=0,x,y;
char buff[32], buff3[50], buff4[50];
float freq,volt,adc,adc_arus; // to store value of frequency
value
unsigned char i=0,dur;
int tegangan;
float av_adc,av_adc1,vout,arus,daya;
char buffer[32], buffer1[10]; // to store the frequency value as
a string to be displayed on lcd
int counter=0,flag=0;
int flow;
float flow_rate;
int max_step=150;
int hitungstep;
char temp[16], temp1[16];
unsigned char wd;
unsigned char dd;
unsigned char mm;
unsigned char yy;
unsigned char s;
unsigned char m;

```

```
unsigned char h;

// External Interrupt 4 service routine
interrupt [EXT_INT4] void ext_int4_isr(void)
{
    // Place your code here
    counter++;
    //counter1++;
}

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
```

```
#define RXC 7
#endif
```

```
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
```

```
// Get a character from the USART1 Receiver
#pragma used+
char getchar1(void)
{
    char status,data;
    while (1)
    {
        while (((status=UCSR1A) & RX_COMPLETE)==0);
        data=UDR1;
        if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
            return data;
    }
}
#pragma used-
```

```
// Write a character to the USART1 Transmitter
#pragma used+
void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}
#pragma used-
```

```
#define _ALTERNATE_PUTCHAR_
```

```

#include <stdio.h>
#define USART0 0           // agar pembacaan tidak acak
#define USART1 1
unsigned char poutput;

void putchar(char c)
{
    switch (poutput)
    {
        case USART0: // the output will be directed to USART0
            while ((UCSR0A & DATA_REGISTER_EMPTY)==0);
            UDR0=c;
            break;

        case USART1: // the output will be directed to USART1
            while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
            UDR1=c;
            break;
    };
}

// Standard Input/Output functions
#include <stdio.h>

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Reinitialize Timer 0 value
    TCNT0=0x8A;
    // Place your code here
    waktu++;
    if (waktu >=120){//kira2 bisa mencapai 1 detik
        //detik++;
        Rpm = counter * 60.0;
        Rpm2 = Rpm * 4.0;
    }
}

```

```

counter = 0;
waktu = 0;
Rpm2 = Rpm2;
}
}

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
    i++;
}

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

// Declare your global variables here
void baca_volt(){
    for(i=0;i<10;i++){
        adc=read_adc(0);
        av_adc=av_adc+adc;
    }
}

```



```

    }
    av_adc=av_adc/10;
    volt=av_adc*106/255;

}

void baca_arus (){
    for (i=0;i<10;i++){
        adc_arus=read_adc(1);
        av_adc1=av_adc1+adc_arus;
    }
    av_adc1=av_adc1/10;
    vout=av_adc1*106/255;
    arus=fabs(vout-2.50)/0.066/1000;
    daya=arus*volt;
}

void baca_ultrasonic(){
    count=0;
    PORTA.1=1;
    delay_us(15);
    PORTA.1=0;
    while(PINA.3==0){ };
    while(PINA.3==1){count++;};
    jrk=count;
    jrk=jrk/100;
    jrk=66.5-jrk;
    jarak = jrk;
    /*if (jrk<0){
        lcd_gotoxy(0,0);
        lcd_putsf("ERROR");}
    else if (jrk>70)
        { lcd_gotoxy(0,0);
        lcd_putsf("ERROR");}
    else {
        lcd_gotoxy(0,0);
        lcd_putsf("Lvl:");

```

```

    //sprintf(buffer,"%d ",jrk);
    ftoa(jrk,2,buffer);
    lcd_puts(buffer);

    lcd_gotoxy(10,0);
    lcd_putsf("Lvl:");
    sprintf(buffer,"%d ",jarak);
    lcd_puts(buffer);
    lcd_putsf(" Cm ");
    delay_ms(100);}*/
}

void baca_flow(){
    TIMSK=0x05;
    TCCR1B=0x07;
    delay_ms(100);
    TCCR1B=0x00;
    TIMSK=0x00;
    dur=TCNT1;
    //flow_rate=0;
    freq = (((dur + i*65536)*600)/4.8)*0.0166;
    TCNT1=0x0000;
    i=0;
    flow_rate=freq;
    flow_rate=flow_rate-1359872.00;
    flow = flow_rate;
    lcd_gotoxy(0,2);
    lcd_putsf("F:");
    //sprintf(buffer,"%d",flow_rate-18350);
    ftoa(flow_rate,2,buffer);
    lcd_puts(buffer);
    lcd_putsf("L/M ");
}

void cecewe_start(){
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);

```

```

b0=0; b1=0; b2=1; b3=0;
delay_ms(5);
b0=0; b1=1; b2=0; b3=0;
delay_ms(5);
b0=1; b1=0; b2=0; b3=0;
delay_ms(5);
b0=0; b1=0; b2=0; b3=1;
delay_ms(5);
b0=0; b1=0; b2=1; b3=0;
delay_ms(5);
b0=0; b1=1; b2=0; b3=0;
delay_ms(5);
b0=1; b1=0; b2=0; b3=0;
delay_ms(5);
}
void cecewe(){ //close
b0=0; b1=0; b2=0; b3=1;
delay_ms(5);
b0=0; b1=0; b2=1; b3=0;
delay_ms(5);
b0=0; b1=1; b2=0; b3=0;
delay_ms(5);
b0=1; b1=0; b2=0; b3=0;
delay_ms(5);
b0=0; b1=0; b2=0; b3=1;
delay_ms(5);
b0=0; b1=0; b2=1; b3=0;
delay_ms(5);
b0=0; b1=1; b2=0; b3=0;
delay_ms(5);
b0=1; b1=0; b2=0; b3=0;
delay_ms(5);
hitungstep--;
persen=(hitungstep*100)/max_step;
if(persen<0){

```

```

    persen=0;
}
/* lcd_gotoxy(13,1);
   sprintf(buffer,"%d ",persen);
   lcd_puts(buffer); */

}

void cewe(){ //open
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    hitungstep++;
    persen=(hitungstep*100)/max_step;
    if(persen>100){
        persen=100;
    }
    /* lcd_gotoxy(13,1);
       sprintf(buffer,"%d ",persen);
       lcd_puts(buffer);*/
}

void stopped(){

```

```

    b3=0; b2=0; b1=0; b0=0;
}
void tampil_persen(){

}

void mov(int kondisi){
    if(kondisi == 1){ //MOV CLOSE
        mov1 = 1;
        mov2 = 0;
        drajat -= 50;
        delay_ms(300);
    }
    if(kondisi == 2){ //MOV OPEN
        mov1 = 0;
        mov2 = 1;
        drajat += 50;
        delay_ms(2400);
    }
    if(kondisi == 3){ //MOV no operation
        mov1 = 1;
        mov2 = 1;
    } }

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTA.1=0;

```

```
PORTA.3=1;  
DDRA.0=1;  
DDRA.1=1;  
DDRA.2=1;  
DDRA.3=0;  
DDRA.4=1;  
DDRA.6=1;
```

```
// Port B initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In  
Func2=In Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTB=0x00;  
DDRB=0xC0;
```

```
// Port C initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In  
Func2=In Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTC=0x00;  
DDRC=0x00;
```

```
// Port D initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In  
Func2=In Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTD=0x00;  
DDRD.4=1;  
DDRD.5=1;
```

```
// Port E initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In  
Func2=In Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTE=0x00;  
DDRE=0x00;
```

```
// Port F initialization  
// Func7=In Func6=In Func5=In Func4=In Func3=In  
Func2=In Func1=In Func0=In  
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T  
PORTF=0x00;  
DDRF=0x00;
```

```
// Port G initialization  
// Func4=In Func3=In Func2=In Func1=In Func0=In  
// State4=T State3=T State2=T State1=T State0=T  
PORTG=0x00;  
DDRG=0x00;
```

```
// Timer/Counter 0 initialization  
// Clock source: System Clock  
// Clock value: 250,000 kHz  
// Mode: Normal top=0xFF  
// OC0 output: Disconnected  
ASSR=0x00;  
TCCR0=0x07;  
TCNT0=0x8A;  
OCR0=0x00;
```

```
// Timer/Counter 1 initialization  
// Clock source: System Clock  
// Clock value: 16000,000 kHz  
// Mode: Normal top=0xFFFF
```

```
// OC1A output: Discon.  
// OC1B output: Discon.  
// OC1C output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer1 Overflow Interrupt: On  
// Input Capture Interrupt: Off  
// Compare A Match Interrupt: Off  
// Compare B Match Interrupt: Off  
// Compare C Match Interrupt: Off  
TCCR1A=0x00;  
TCCR1B=0x01;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;  
OCR1CH=0x00;  
OCR1CL=0x00;  
  
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected  
TCCR2=0x00;  
TCNT2=0x00;  
OCR2=0x00;  
  
// Timer/Counter 3 initialization  
// Clock source: System Clock  
// Clock value: Timer3 Stopped
```



```
// Mode: Normal top=0xFFFF
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: On
// INT4 Mode: Rising Edge
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
```

```
EICRB=0x03;  
EIMSK=0x10;  
EIFR=0x10;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization  
TIMSK=0x05;
```

```
ETIMSK=0x00;
```

```
// USART0 initialization  
// Communication Parameters: 8 Data, 1 Stop, No Parity  
// USART0 Receiver: On  
// USART0 Transmitter: On  
// USART0 Mode: Asynchronous  
// USART0 Baud Rate: 9600  
UCSR0A=0x00;  
UCSR0B=0x98;  
UCSR0C=0x06;  
UBRR0H=0x00;  
UBRR0L=0x67;
```

```
// USART1 initialization  
UCSR1A=0x00;  
UCSR1B=0x18;  
UCSR1C=0x06;  
UBRR1H=0x00;  
UBRR1L=0x67;
```

```
// Analog Comparator initialization  
// Analog Comparator: Off  
// Analog Comparator Input Capture by Timer/Counter 1: Off  
ACSR=0x80;  
SFIOR=0x00;
```

```
// ADC initialization
```

```

// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AVCC pin
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

i2c_init();
rtc_init(0,0,0);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 3
// D5 - PORTC Bit 4
// D6 - PORTC Bit 5
// D7 - PORTC Bit 6
// Characters/line: 20
lcd_init(20);

// Global enable interrupts
#asm("sei")

```

```

/*****setup time and date*****/
h=22;m=28;s=00;
rtc_set_time(h,m,s);

dd=16;mm=07;yy=18;
rtc_set_date(wd,dd,mm,yy);
/*****

```

```

pompa1=1;
pompa2=0;
mov1=0;
mov2=1;
lcd_gotoxy(0,0);
lcd_putsf("  WELCOME TO  ");
lcd_gotoxy(0,1);
lcd_putsf("  PLTMH PLANT  ");
lcd_gotoxy(0,2);
lcd_putsf("    2018    ");
for(i=0;i<160;i++){
    cecewe_start();
}

```

```

delay_ms(1500);
lcd_clear();
while (1)
{
    // Place your code here
    start:
    baca_ultrasonic();
    baca_flow();
    baca_volt();
    baca_arus();

    lcd_gotoxy(0,0);
    lcd_putsf("L:");
}

```

```
sprintf(buff, "L:%d cm ", jarak);  
ftoa(jrk, 2, buffer);  
lcd_puts(buffer);  
lcd_putsf("cm");
```

```
lcd_gotoxy(11, 0);  
lcd_putsf("V:");  
ftoa(volt, 2, buff);  
lcd_puts(buff);  
lcd_putsf("V");
```

```
sprintf(temp1, "N:%d RPM", Rpm);  
lcd_gotoxy(0, 1);  
lcd_puts(temp1);
```

```
lcd_gotoxy(11, 1);  
lcd_putsf("I:");  
ftoa(arus, 2, buff);  
lcd_puts(buff);  
lcd_putsf("A");
```

```
lcd_gotoxy(11, 2);  
lcd_putsf("P:");  
ftoa(daya, 2, buff);  
lcd_puts(buff);  
lcd_putsf("W");
```

```
lcd_gotoxy(11, 3);  
sprintf(buffer, "MOV2:%d", persen);  
lcd_puts(buffer);  
lcd_putsf("%");
```

```
lcd_gotoxy(0, 3);  
sprintf(buffer1, "N2:%d Rpm", Rpm2);  
lcd_puts(buffer1);
```

```

    lcd_putsf("%");

    //=====data logger=====
    sprintf(buff3, "%02u:%02u:%02u/%02u:%02u:%02u ;
%.2f cm ; %d RPM ; %.2f lpm ; %.2f V\r",dd,mm,yy,s,m,h,
jrk, Rpm, flow_rate, volt);
    poutput = USART1;
    puts(buff3);

//=====
=====HMI=====
=====

    // matikan salah satu kodingan, karena dalam kodingan ini
    kedua HMI belum diintegrasikan

    // hmi baru
    sprintf(buff4, "%.2f;%d\r", volt, Rpm);
    poutput = USART0;
    puts(buff4);*/
    poutput = USART0;

    printf("%.2f;%d;%d;%.2f;%.2f\r",jrk,Rpm,Rpm2,flow_rate,v
olt);

    // hmi lama
    //sprintf(buff4, "JARAK: %.2f; RPM: %d; FLOW: %.2f;
TEGANGAN: %.2f\r", jrk, Rpm, flow_rate, volt);
    //poutput = USART0;
    //puts(buff4);

//=====
=====
=====

    // delay_ms(300);

```

```

if(volt<38){
    ceuwe();
    tampil_persen();
}
else if(volt>38.5){
    ceceuwe();
    tampil_persen();
}
else {
    stopped();
    tampil_persen();
}
}
acuan_mov= jrk - 35;

```

```

    if(jarak==35 && flag==0){
pompa1 =0; //pompa1
pompa2 =0; //pompa2
mov1=0;
delay_ms(1500);
mov1=1;
flag=1;
delay_ms(1000);
goto start;
}

```

```

if(jarak>45 && flag==1){
    pompa1=1;
    pompa2=1;

}
else if(jarak>35 && jarak<45 && flag==1){
    pompa1=0;
    pompa2=0;
}

```

```

mov2=0; //
mov1=1;
drajat=0;
x=1;
y=0;
}
else if(jarak==35 && flag==1 && x==1){
    pompa1=0;
    pompa2=0;
    if(drajat < 50){
        mov(2); //MOV OPEN
        mov(3);
    }
}
else if(jarak==35 && flag==1 && y==1){
    pompa1=0;
    pompa2=0;
    if(drajat > 50){
        mov(1); //MOV OPEN
        mov(3);
    }
}
}
else if(jarak>25 && jarak<35 && flag==1){
    pompa1=0;
    pompa2=0;
    mov1=1; //
    mov2=0;
    drajat=100;
    x=1;
    y=0;
}
else if(jarak<25 && flag==1){
    pompa1=1;
    pompa2=1;

```



```
    } //delay_ms(250);  
  }  
}
```



## LAMPIRAN B

### (DATA SHEET SYSTEM ATMEGA128)

#### Features

- High-performance, Low-power Atmel®AVR®8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 128Kbytes of In-System Self-programmable Flash program memory
  - 4Kbytes EEPROM
  - 4Kbytes Internal SRAM
  - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Up to 64Kbytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
  - Output Compare Modulator
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V ATmega128L
  - 4.5 - 5.5V ATmega128
- Speed Grades
  - 0 - 8MHz ATmega128L
  - 0 - 16MHz ATmega128



**8-bit Atmel  
Microcontroller  
with 128KBytes  
In-System  
Programmable  
Flash**

**ATmega128  
ATmega128L**

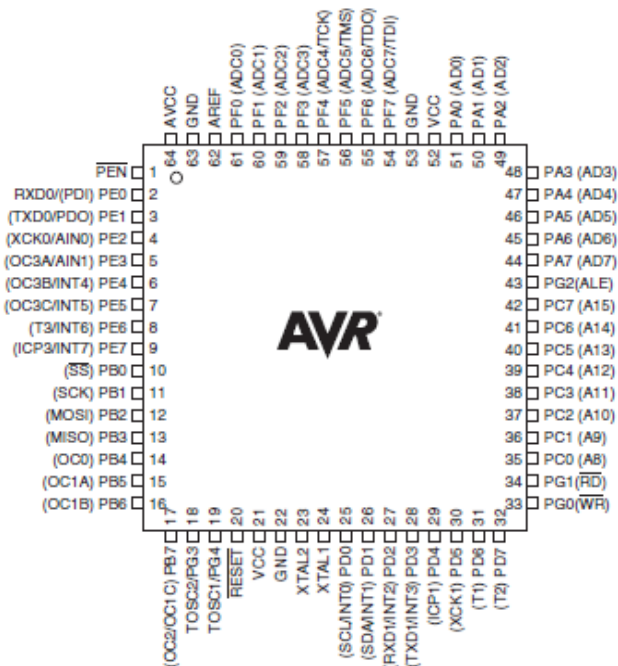
**Summary**

Rev. 246725-AVR-0011



## Pin Configurations

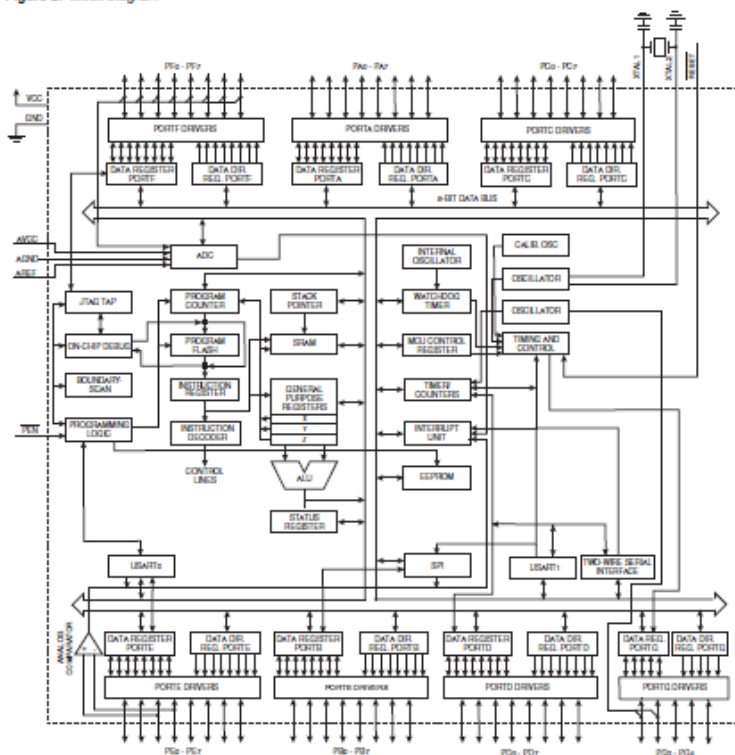
Figure 1. Pinout ATmega128



Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

## Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

**Block Diagram**
**Figure 2. Block Diagram**


The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughput up to ten times faster than conventional CISC microcontrollers.

The ATmega128 provides the following features: 128Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 4Kbytes SRAM, 53 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128 device is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## ATmega103 and ATmega128 Compatibility

The ATmega128 is a highly complex microcontroller where the number of I/O locations supersedes the 64 I/O locations reserved in the AVR instruction set. To ensure backward compatibility with the ATmega103, all I/O locations present in ATmega103 have the same location in ATmega128. Most additional I/O locations are added in an Extended I/O space starting from \$60 to \$FF, (i.e., in the ATmega103 internal RAM space). These locations can be reached by using LD/ LDS/ LDD and ST/ STS/ STD instructions only, not by using IN and OUT instructions. The relocation of the internal RAM space may still be a problem for ATmega103 users. Also, the increased number of interrupt vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega103 compatibility mode can be selected by programming the fuse M103C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega103. Also, the Extended Interrupt vectors are removed.

## ATmega103 Compatibility Mode

The ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

By programming the M103C fuse, the Atmel® ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, Asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16-bit Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port C is output only.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.

In addition, there are some other minor differences to make it more compatible to ATmega103:

- Only EXTRF and PORF exists in MCUCSR.
- Timed sequence not required for Watchdog Time-out change.
- External Interrupt pins 3 - 0 serve as level interrupt only.
- USART has no FIFO buffer, so data overrun comes earlier.

Unused I/O bits in ATmega103 should be written to 0 to ensure same operation in ATmega128.

## Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.  Port A also serves the functions of various special features of the ATmega128 as listed on <a href="#">page 72</a> .
Port B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.  Port B also serves the functions of various special features of the ATmega128 as listed on <a href="#">page 73</a> .

## Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the Atmel® AVR<sup>®</sup> ATmega128 as listed on [page 76](#). In ATmega103 compatibility mode, Port C is output only, and the port C pins are not tri-stated when a reset condition becomes active.

**Note:** The ATmega128 is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.

## Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega128 as listed on [page 77](#).

## Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega128 as listed on [page 80](#).

## Port F (PF7..PF0)

Port F serves as the analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.

The TDO pin is tri-stated unless TAP states that shift out data are entered.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input Port only.

## Port G (PG4..PG0)

Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features.

The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.



In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32KHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

<b>RESET</b>	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in <a href="#">Table 19 on page 50</a> . Shorter pulses are not guaranteed to generate a reset.
<b>XTAL1</b>	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
<b>XTAL2</b>	Output from the inverting Oscillator amplifier.
<b>AVCC</b>	AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to $V_{CC}$ through a low-pass filter.
<b>AREF</b>	AREF is the analog reference pin for the A/D Converter.
<b>PEN</b>	PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. PEN has no function during normal operation.

**Resources**

A comprehensive set of development tools, application notes, and datasheets are available for download on <http://www.atmel.com/avr>.

**Data Retention**

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C

**About Code Examples**

This datasheet contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBR", "SBR", and "CBR".

**Capacitive touch sensing**

The Atmel QTouch Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: [www.atmel.com/qtouchlibrary](http://www.atmel.com/qtouchlibrary). For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from the Atmel website.

## Instruction Set Summary

Mnemonic	Operands	Description	Operation	Flags	#Cycles
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H	1
ADW	Rd, K	Add Immediate to Word	$IndirectRd \leftarrow IndirectRd + K$	Z, C, N, V, H	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H	1
SUB	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z, C, N, V, H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z, C, N, V, H	1
SBC	Rd, K	Subtract with Carry Constant from Reg	$Rd \leftarrow Rd - K - C$	Z, C, N, V, H	1
SDW	Rd, K	Subtract Immediate from Word	$IndirectRd \leftarrow IndirectRd - K$	Z, C, N, V, H	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \wedge Rr$	Z, N, V	1
AND	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \wedge K$	Z, N, V	1
ORI	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z, N, V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z, N, V	1
LCR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z, N, V	1
COM	Rd	One's Complement	$Rd \leftarrow \sim Rd$	Z, C, N, V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \sim Rd + 1$	Z, C, N, V, H	1
SEI	Rd, Rr	Set (Stk) in Register	$Rd \leftarrow Rd \vee K$	Z, N, V	1
CLR	Rd, Rr	Clear (Stk) in Register	$Rd \leftarrow Rd \wedge \sim K$	Z, N, V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z, N, V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z, N, V	1
TEST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \wedge Rd$	Z, N, V	1
CLP	Rd	Clear Register	$Rd \leftarrow Rd \wedge Rd$	Z, N, V	1
SRP	Rd	Set Register	$Rd \leftarrow \sim Rd$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R16 \leftarrow Rd \times Rr$	Z, C	2
MULS	Rd, Rr	Multiply Signed	$R16 \leftarrow Rd \times Rr$	Z, C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R16 \leftarrow Rd \times Rr$	Z, C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R16 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R16 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R16 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
<b>BRANCH INSTRUCTIONS</b>					
JMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
LJMP	k	Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
ICALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL	k	Indirect Call to (Z)	$PC \leftarrow Z$	None	3
LCALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	1	4
CPSE	Rd, Rr	Compare, Skip if Equal	$Rd \wedge Rr ? PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
CPI	Rd, Rr	Compare	$Rd \wedge Rr$	Z, N, V, C, H	1
CPD	Rd, Rr	Compare with Carry	$Rd \wedge Rr - C$	Z, N, V, C, H	1
CPDI	Rd, K	Compare Register with Immediate	$Rd \wedge K$	Z, N, V, C, H	1
SEDFR	Rd, Rr	Set if (Rd) Register Cleared	$Rd \wedge \sim Rr ? PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SEDFR	Rd, Rr	Set if (Rd) Register is Set	$Rd \wedge Rr ? PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SEDFR	Rd, Rr	Set if (Rd) Register Cleared	$Rd \wedge \sim Rr ? PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SEDFR	Rd, Rr	Set if (Rd) Register is Set	$Rd \wedge Rr ? PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SEDFR	k, k	Search if (Rd) Flag Set	$Rd \wedge (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k, k	Search if (Rd) Flag Cleared	$Rd \wedge \sim (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Equal	$Rd \wedge (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Not Equal	$Rd \wedge \sim (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Carry Set	$Rd \wedge C ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Carry Cleared	$Rd \wedge \sim C ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Set or Higher	$Rd \wedge (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Lower	$Rd \wedge \sim (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Minus	$Rd \wedge M ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Plus	$Rd \wedge \sim M ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Greater or Equal, Signed	$Rd \wedge (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Less Than, Signed	$Rd \wedge \sim (Rd \wedge k) ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Carry Flag Set	$Rd \wedge C ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Carry Flag Cleared	$Rd \wedge \sim C ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Flag Set	$Rd \wedge F ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Flag Cleared	$Rd \wedge \sim F ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Overflow Flag is Set	$Rd \wedge O ? PC \leftarrow PC + k + 1$	None	1/2
SEDFR	k	Search if Overflow Flag is Cleared	$Rd \wedge \sim O ? PC \leftarrow PC + k + 1$	None	1/2

# Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	#Cycles
SFIS	s	Branch if Interrupt Flag Set	$B(1 = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
SFID	s	Branch if Interrupt Flag Cleared	$B(1 = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	dst, sr	Move between Registers	$dst \leftarrow sr$	None	1
MOVW	dst, sr	Copy Register Word	$dst \leftarrow 16b \text{ of } sr$	None	1
LDI	dst, k	Load Immediate	$dst \leftarrow k$	None	1
LDD	dst, sr	Load Indirect	$dst \leftarrow (sr)$	None	2
LDD	dst, X+	Load Indirect and Post-Inc.	$dst \leftarrow (X); X \leftarrow X + 1$	None	2
LDD	dst, -X	Load Indirect and Pre-Dec.	$X \leftarrow X - 1; dst \leftarrow (X)$	None	2
LDD	dst, Y	Load Indirect	$dst \leftarrow (Y)$	None	2
LDD	dst, Y+	Load Indirect and Post-Inc.	$dst \leftarrow (Y); Y \leftarrow Y + 1$	None	2
LDD	dst, Y-	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1; dst \leftarrow (Y)$	None	2
LDD	dst, Z	Load Indirect	$dst \leftarrow (Z)$	None	2
LDD	dst, Z+	Load Indirect and Post-Inc.	$dst \leftarrow (Z); Z \leftarrow Z + 1$	None	2
LDD	dst, Z-	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1; dst \leftarrow (Z)$	None	2
LDD	dst, Z+6	Load Indirect with Displacement	$dst \leftarrow (Z + 6)$	None	2
LDD	dst, X	Load Direct from SRAM	$dst \leftarrow (X)$	None	2
ST	X, sr	Store Indirect	$(X) \leftarrow sr$	None	2
ST	X+, sr	Store Indirect and Post-Inc.	$(X) \leftarrow sr; X \leftarrow X + 1$	None	2
ST	-X, sr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1; (X) \leftarrow sr$	None	2
ST	Y, sr	Store Indirect	$(Y) \leftarrow sr$	None	2
ST	Y+, sr	Store Indirect and Post-Inc.	$(Y) \leftarrow sr; Y \leftarrow Y + 1$	None	2
ST	-Y, sr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1; (Y) \leftarrow sr$	None	2
STD	Y+6, sr	Store Indirect with Displacement	$(Y + 6) \leftarrow sr$	None	2
ST	Z, sr	Store Indirect	$(Z) \leftarrow sr$	None	2
ST	Z+, sr	Store Indirect and Post-Inc.	$(Z) \leftarrow sr; Z \leftarrow Z + 1$	None	2
ST	Z-, sr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1; (Z) \leftarrow sr$	None	2
STD	Z+6, sr	Store Indirect with Displacement	$(Z + 6) \leftarrow sr$	None	2
STB	X, sr	Store Direct to SRAM	$(X) \leftarrow sr$	None	2
LPM		Load Program Memory	$rd \leftarrow (Z)$	None	3
LPM	rd, Z	Load Program Memory	$rd \leftarrow (Z)$	None	3
LPM	rd, Z+	Load Program Memory and Post-Inc.	$rd \leftarrow (Z); Z \leftarrow Z + 1$	None	3
ELPM	rd, Z	Extended Load Program Memory	$rd \leftarrow (RAM2Z)$	None	3
ELPM	rd, Z+	Extended Load Program Memory and Post-Inc.	$rd \leftarrow (RAM2Z); Z \leftarrow Z + 1$	None	3
SPM		Store Program Memory	$(Z) \leftarrow r16$	None	—
IN	sr, D	In Port	$sr \leftarrow P$	None	1
OUT	D, sr	Out Port	$P \leftarrow sr$	None	1
PUSH	sr	Push Register on Stack	$STACK \leftarrow sr$	None	2
POP	sr	Pop Register from Stack	$sr \leftarrow STACK$	None	2
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
SBIC	P/D	Set Bit in I/O Register	$IOIF(D) \leftarrow 1$	None	2
SBIS	P/D	Clear Bit in I/O Register	$IOIF(D) \leftarrow 0$	None	2
LSD	dst	Logical Shift Left	$dst(n+1) \leftarrow dst(n); dst(0) \leftarrow 0$	Z, C, N, V	1
LSD	dst	Logical Shift Right	$dst(n) \leftarrow dst(n+1); dst(7) \leftarrow 0$	Z, C, N, V	1
RCL	dst	Rotate Left Through Carry	$dst(n) \leftarrow C; dst(n+1) \leftarrow dst(n); dst(7) \leftarrow C$	Z, C, N, V	1
RCR	dst	Rotate Right Through Carry	$dst(7) \leftarrow C; dst(6) \leftarrow dst(7); dst(n) \leftarrow C$	Z, C, N, V	1
ASR	dst	Arithmetic Shift Right	$dst(n) \leftarrow dst(n+1); dst(0) \leftarrow 0$	Z, C, N, V	1
SWAP	dst	Swap Nibbles	$dst(3) \leftarrow dst(7); dst(7) \leftarrow dst(3)$	None	1
SEI	s	Flag Set	$SREG \leftarrow 1$	SF, Cof	1
CLI	s	Flag Clear	$SREG \leftarrow 0$	SF, Cof	1
SBIC	sr, b	SB Set from Register to 1	$1 \leftarrow dst(b)$	T	1
SBIS	sr, b	SB Set from Register to 0	$dst(b) \leftarrow 1$	None	1
SBIC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SEIS		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLIS		Clear Signed Test Flag	$S \leftarrow 0$	S	1

**Instruction Set Summary (Continued)**

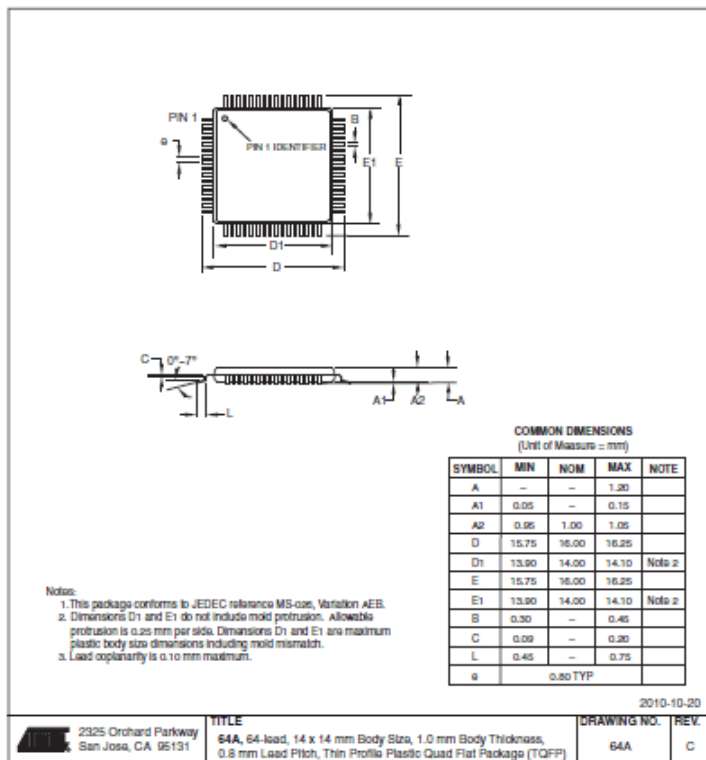
Mnemonic	Operands	Description	Operation	Flags	#Cycles
SEV		Set Zero Complement Overflow	$Y \leftarrow 1$	Y	1
CLV		Clear Zero Complement Overflow	$Y \leftarrow 0$	Y	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEI		Set Interrupt Enable in SREG	$IE \leftarrow 1$	IE	1
CLI		Clear Interrupt Enable in SREG	$IE \leftarrow 0$	IE	1
<b>MCU CONTROL INSTRUCTIONS</b>					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific device for Sleep function)	None	1
WDR		Watchdog Reset	(see specific device for WDR/Timer)	None	1
BSCSR		Reset	For On-chip Debug Only	None	N/A

## Ordering Information

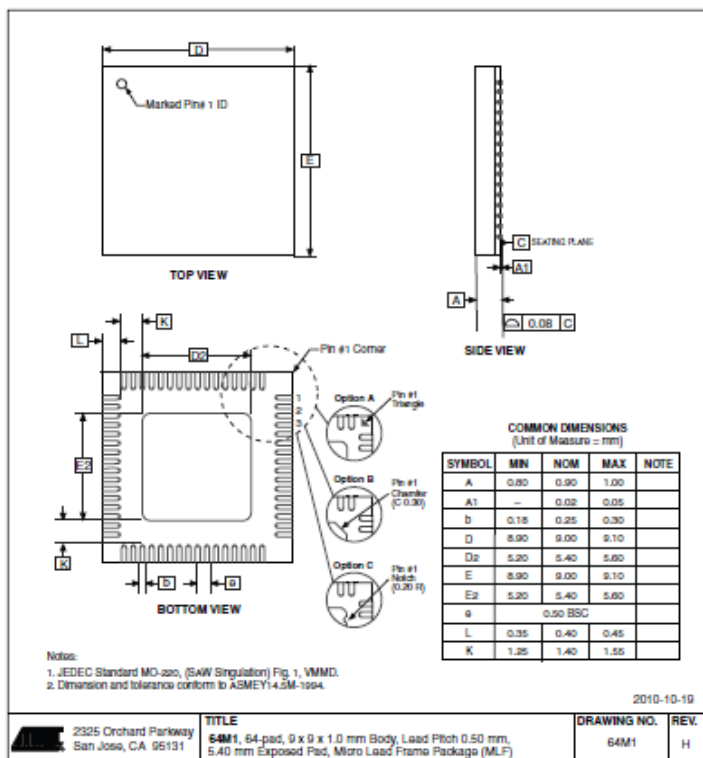
Speed (MHz)	Power Supply	Ordering Code <sup>(1)</sup>	Package <sup>(2)</sup>	Operation Range
8	2.7 – 5.5V	ATmega128L-8AJ ATmega128L-8AJR <sup>(3)</sup> ATmega128L-8MU ATmega128L-8MUR <sup>(3)</sup>	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)
16	4.5 – 5.5V	ATmega128-16AJ ATmega128-16AJR <sup>(3)</sup> ATmega128-16MU ATmega128-16MUR <sup>(3)</sup>	64A 64A 64M1 64M1	
8	3.0 – 5.5V	ATmega128L-8AN ATmega128L-8ANR <sup>(3)</sup> ATmega128L-8MN ATmega128L-8MNR <sup>(3)</sup>	64A 64A 64M1 64M1	Extended (-40°C to 105°C)
16	4.5 – 5.5V	ATmega128-16AN ATmega128-16ANR <sup>(3)</sup> ATmega128-16MN ATmega128-16MNR <sup>(3)</sup>	64A 64A 64M1 64M1	

- Notes: 1. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.  
 2. The device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.  
 3. Tape and Reel

Package Type	
64A	64-lead, 14 x 14 x 1.0mm, Thin Profile Plastic Quad Flat Package (TQFP)
64M1	64-pad, 9 x 9 x 1.0mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

**Packaging Information**
**64A**


64M1





**LAMPIRAN C**  
**(DATA SHEET WATER FLOW SENSOR TYPE FS400A)**



**Gambar C.** Water flow sensor FS400A 1”

Spesifikasi G 1 inch water flow sensor :

- Rentang tegangan operasional antara 5 ~ 24 Volt DC
- Maksimum arus operasional sebesar 15 mA (pada 5VDC)
- Maksimum suhu air 120°C
- Maksimum suhu operasional 80°C
- Rentang kelembapan operasional antara 35% ~ 90% RH
- Maksimum tekanan air 1,2 MPa
- Diameter thread inlet/outlet 1½ (~25 mm)
- Range flow : 1 liter/ menit ; V 60 liter/menit
- Calculation Formula :  $F(Hz) = 4.8 * Q(L/min)$

Catatan:

- Ketika bahan magnetik dekat dengan sensor ,karakteristiknya dapat bervariasi .
- Untuk menghindari puing-puing partikel , sensor harus dipasang setelah filter .
- Aliran instalasi sensor harus menghindari getaran yang kuat dan gemetar lingkungan , sehingga tidak mempengaruhi akurasi pengukuran sensor .

## BIODATA PENULIS



Penulis bernama Nila Dia Citra Effendy, lahir di Lamongan pada tanggal 27 April 1998, merupakan anak pertama dari empat bersaudara. Penulis telah menempuh pendidikan formal di SD Islamiyah 1 Ternate, SMPN 2 Laren, dan SMA Muhammadiyah 1 Babat. Penulis mengikuti seleksi Masuk D3 ITS Pada tahun 2015, dan diterima di Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh

Nopember Surabaya.

Penulis pernah mengikuti kegiatan organisasi dan pelatihan selama menjadi mahasiswa. Penulis pernah menjadi staf Kaderisasi JMMI 2016/2017, staf Keputrian LDJ Fusi Ulul Albab 2016/2017 dan sekretaris UKM Jujitsu 2016/2017. Penulis pernah melakukan kerja praktik di Pembangkitan Jawa Bali (PJB) UP Paiton, Jawa Timur dengan judul Pengaturan *Governor* Dalam Menjaga Kestabilan Putaran Turbin Uap di PLTU PJB Paiton Unit 1&2, selama kurang lebih satu bulan. Pada masa akhir studi Penulis mengerjakan tugas akhir di Laboratorium Mikrokontroler dan Kalibrasi dengan judul Sistem Pengendalian Tegangan Dengan Pengaturan Laju Aliran Nosel Pada PLTMH Skala Laboratorium dibawah bimbingan dan arahan Bapak Hendra Cordova, S.T,M.T dan Ibu Sefi Novendra Patrialova, S.Si,M.T. Penulis dapat dihubungi di 085859456880 atau effendynila@gmail.com.